The magazine for Sinclair users and Timex/Sinclair users $3.95

SYNC

January/February 1984 Volume 4, Number 1

THEME SECTION: SYNC IN THE HOME OFFICE

Check Your Tax Shelter • Make a Speadsheet • Hatch Your Nest Egg • Control Your Home • PROGRAMMING: Split 'n Save • Chaining Programs • MACHINE LANGUAGE: Bit by Bit • Search and Replace Routines • GAMES: Highway Robbery • REVIEWS: Word Processing Packages • Keyboards

The magazine for Sinclair users and TIMEX/Sinclair users

SYNC

January/February 1984 — Volume 4, Number 1

Staff

Founder/Editor-in-Chief	David Ahl
Managing Editor	Paul Grosjean
Contributing Editor	David Ornstein
Art Editor	Diana Negri Rudio
Typesetting	Karen Brown
Advertising Sales Manager	Wayne Stephens
Advertising Coordinator	Julie Henderson

Ziff-Davis Publishing Company
Consumer Computer and Electronics Division

President	Larry Sporn
Vice President/ Marketing	J. Scott Briggs
Vice President/ Circulation	Carole Mandel
Vice President/ General Manager	Eileen G. Markowitz
Creative Director	Peter J. Blank
Promotion Manager	Ronni Sonnenberg

Volume 4, Number 1

SYNC (USPS 585-490 ISSN 0279-5701) is published bi-monthly by Ahl Computing Inc., a subsidiary of Ziff-Davis Publishing Company. David Ahl, President; Elizabeth B. Staples, Vice-President; Selwyn Taubman, Treasurer; Bertram A. Abrams, Secretary. 39 E. Hanover Ave., Morris Plains, NJ 07950. Second class postage paid at New York, NY 10001 and at additional mailing offices.

Subscription rates: USA: One year (6 issues) $19.97; two years (12 issues) $36.97; three years (18 issues) $49.97. Canada and other foreign $4 per year additional.

Subscriptions: For all inquiries concerning subscriptions, new orders, renewals, change of address (include the name and old ZIP or mailing label), problems, etc., write to: SYNC Magazine, PO Box 2939, Boulder, CO 80302.

Advertising: For advertising information contact Wayne Stephens, *SYNC* Advertising Sales Manager, Ziff-Davis Publishing Company, One Park Ave., New York, NY 10016 (phone 212/725-4216).

All other correspondence: Address to *SYNC*, 39 E. Hanover Ave., Morris Plains, NJ 07950. In U.K. to *SYNC*, 10 Bishops Way, Sutton Coldfield, W. Midlands B74 4XU.

Postmaster: Send address changes to: *SYNC*, PO Box 789-M, Morristown, NJ 07960.

Cover by **Bob Aiese**

$59.95 VOICE SYNTHESIZER

- Now you can purchase the Zebra Talker unlimited vocabulary voice synthesizer for only $59 95
- The Zebra Talker is TS1000 and ZX 81 compatible
- The Zebra-Talker voice software (requires 16K) will help you create whatever you want to say
- The Zebra Talker uses the famous VOTRAX SC01 phoneme synthesizer with programmable pitch levels. It produces all the phonemes required for English speech
- It is so simple to use you can add voice to any program in just minutes
- There are hundreds of applications in education, robotics, speech therapy monitoring, games prompting and more
- Expansion connector for "Piggy Back" expandability
- You can plug the Zebra-Talker's output into any audio system, or order our #C145 Amplifier/Speaker module with volume control for just $12 50
- Includes assembled & tested Zebra Talker, instruction manual, and software on cassette

SPEAKER / AMPLIFIER
ORDER C145 $12 50

Zebra-Talker
FOR THE TS1000 & ZX-81
ORDER C141 $59 95

BLIPPO

- TS1000 ZX81 Sound effects generator
- Add amazing sound effects to your games
- Easy to program in BASIC
- Wide frequency range
- Features bus expansion connector
- Assembled & Tested with Demo software

C122 BLIPPO WITHOUT SPEAKER $16 95
C124 Attractive Speaker Module $5 00

GREEN SCREEN

The Green Screen is a specially molded plastic optical filter which relieves eye strain by enhancing character contrast and by eliminating the components of white light which cause eye fatigue. The Green Screen will fit any 12" TV or monitor and can easily be trimmed with a pair of scissors to fit any smaller size

ORDER C140 $12 95

JOYSTICK ADAPTOR

The ZEBRA Joystick Adaptor allows you to connect any Atari compatible joystick to your ZX81 or TS1000. It comes assembled and tested with simple instructions to make joystick versions of most action games. Simply plugs into the back of your computer and features an expansion connector for piggybacking. Use your own Atari compatible joystick or add $6.50 for a genuine Atari Joystick. **FREE PATCHKIT #1 shows patches for TIMEX Flight Simulator and other popular games.**

C120 ZEBRA JOYSTICK ADAPTOR $16 95
C121 ATARI JOYSTICK $ 6.50
C115 8 JOYSTICK DEMO GAMES CASSETTE $ 9 95

ZEBRA LIGHT PEN

- Fun to experiment in BASIC
- Complete package includes
  - Light Pen
  - Computer Interface Module
  - Demo Program Listings
- Assembled & Tested

ORDER C109 $24 95

ALL ZEBRA PRODUCTS HAVE OUR 15-DAY MONEY BACK GUARANTEE

The **Timex Sinclair Printer** for your Timex Sinclair 1000 or Sinclair ZX81 prints 32 columns wide two lines per second with full graphics and text capability. The TS2040 prints silently in two modes 1) full screen printing with single key copy command and 2) program controlled printing allowing for custom control of the printing format. The TS2040 connects easily to your computer has simple ON/OFF controls comes with one roll of paper easy to understand instructions and a 90-day limited warranty

Order Zebra Cat. #C144 $84.95

SUGGESTED RETAIL
$99.95
ONLY
$84.95
Timex Sinclair 2040 Printer

Printer Paper THERMAL PAPER FOR THE TS2040 SIX ROLL PACK #C143 $11 90

KEYBOARD BEEPER

Experience for yourself how much the Keyboard Beeper improves the use of your TIMEX keyboard. Simply plug it into your computer's expansion connector and hear a beep whenever a character is accepted by the computer's keyboard input routine. No software is required. Features expansion connector for RAM etc

BONUS ! You can trigger the beep from Basic. Great for program prompts timers

ORDER #C133 $19 95

THE BEST BOOKS

C134 ELECTRONICALLY SPEAKING COMPUTER SPEECH GENERATION by John Cater 230 pages. Very enjoyable reading Useful & up-to-date $14.95
C135 LEARNING TIMEX SINCLAIR BASIC by David Lein 350 pages. Great for beginners $14 95
C103 NOT JUST 30 PROGRAMS FOR THE ZX81 Interesting programs with explanations $ 9.95
C162 BYTING DEEPER INTO YOUR TS1000 Excellent presentation with 37 tutorial programs $12 95
C105 UNDERSTANDING YOUR ZX81 ROM by Ian Logan. Good study of ZX81 & assembly language $14 95
C108 THE COMPLETE TS1000/ZX81 ROM DISASSEMBLY by Ian Logan. A must for advanced users $16 95
C106 MACHINE LANGUAGE MADE SIMPLE FOR YOUR SINCLAIR & TIMES TS1000. $14.95
C107 THE INS & OUTS OF THE TS1000 & ZX81. Excellent hardware manual $12.95

EXPERIMENTING?

C111 UNIVERSAL PATTERN FOR IC'S CAPS, ETC.
C112 PIGGY BACK RAM OR OTHER PERIPHERALS
C110 ALL SIGNALS LABELED

C110 Keyed & Labeled ZX Connector $5 95
C111 Universal Prototyping Board $9 95
C112 Expansion Connector $2 50

ZEBRA SYSTEMS, INC.

78-06 Jamaica Avenue, Woodhaven, New York 11421 (212) 296-2385

WE ACCEPT VISA, MASTERCARD
CHECKS, M.O.
N.Y. RESIDENTS ADD SALES TAX
DEALER DISCOUNTS AVAILABLE

CIRCLE 69 ON READER SERVICE CARD

letters

RUNning with Reduced RAM

Dear Editor

I would like to suggest a different method of lowering the RAM under which a program is SAVEd ("RUNning with Reduced RAM," *SYNC* 3 4), especially if variables prevent it from being lowered. LOAD the program into 16K and CLEAR the variables. Reset RAMTOP by POKEing locations 16388 and 16389. Then SAVE the program without the recorder running. Finally SAVE the program to tape under the reduced RAM.

Ed Haymore
11055 Samedra St
Cupertino, CA 95014

Robert Hartung—Another suggestion came from John Oliger, who has frequently written technical articles for **Syntax/SQ:** *Set up a new minimal display file before SAVE by POKEing the RAMTOP address without using NEW, then CLS, then SAVE. By POKEing 16389 with a value from 68 through 76, without NEW, and either before or after a program is listed, and after a CLS, SAVEing time will be reduced by 15 seconds.*

This method will SAVE the listing along with any defined variables or arrays which exist above the pseudo-RAMTOP, but such a SAVE can only be LOADed back into a full 16K RAM, or at least a RAMTOP definition large enough to hold all of it, including defined variables and arrays. This is no problem if less than 3.25K RAM is required, but, when a listing requiring more than 3.25K RAM is SAVEd with the display file collapsed, it can take up to 45 seconds upon reLOADing for the display file to expand and the other system files to relocate before the program can RUN. This is also true if the display file is collapsed by a 22-SCROLL loop.

To be sure, NEW is not needed to reset the RAMTOP registers if the only purpose is to establish a minimal display file before a SAVE, but there are additional advantages to the procedure. Moving RAMTOP below 3.25K (with or without NEW) reduces the time required by CLS. This speeds up Basic moving graphics routines that use CLS repeatedly. Furthermore, using NEW when resetting the RAMTOP registers prevents overwriting any data or routines (e.g., a bytes remaining routine) in the upper memory area by a lengthy programs, large DIM definitions, or system files. Resetting RAMTOP with NEW to 1K or 2K makes it unnecessary to remove the 16K RAM pack when writing programs for the unexpanded machine.

So I believe the original premise of my article is still valid.

Brick Busters

Dear Editor:

I would like to suggest some improvements to Paul Thomson's "Brick Buster" (*SYNC* 3 5). It is a fine program while working in assembler, but it is simply too slow when it switches to Basic.

As an alternative when it computes the score, delete lines 220-250 and then enter these lines:

```
  5 LET D=PEEK 16396+256*PEEK 16397
220 LET S$=CHR$(PEEK(D+2))+CHR$(PEEK(D+3))+CHR$(PEEK(D+4))+CHR$(PEEK(D+5))
230 LET SCORE=VAL S$
```

This modification will be at least 50 times faster, and (may I say it?) more elegant.

The field can be drawn instantaneously by adding these lines.

```
   2 REM (52 characters)
1000 FOR X=16790 TO 16841
1010 INPUT A
1020 POKE X,A
1030 PRINT PEEK X;" ";
1040 NEXT X
```

Press RUN 1000 and ENTER. Then enter the following numbers: 42,12,64,35,54,131,35,6,30,54,131,35,16,251,54,4,35,35,6,7,54,133,35,14,30,54,8,35,13,32,250,54,5,35,35,16,239,17,31,0,6,13,54,133,25,54,5,35,35,16,247,201.

Delete lines 70-130 and 1000-1040 and add

```
100 RAND USR 16790
110 PRINT AT 0,1;"0000"
```

If you do not like the black blob used as the ball, just POKE the value 52 to locations 16589, 16663, 16681, and 16708.

SAVE and then RUN and have FUN.

I enjoy your publication very much, and I am especially interested in articles on machine language. Keep up the good work.

Salvatore Cimmino
4380 Denis Papin
St-Leonard, P.Q.
Canada H1L 1R9

Help

Dear Editor:

Does anybody know the American equivalent to the ZTX-750 or ZTX-752 transistor used in the 16K RAM pack as the heart of the built-in oscillator/power converter for producing the +12V and -5V required for the memory chips in the RAM pack?

J. L. Peeler
Apdo 41 Pavas, San Jose 1200
Costa Rica

Dear Editor

I have noticed that most other computers have a POKE command that disables their break keys. Is there any such command for the TS1000? If so, what is it? Also, is there a program for the TS1000 that lets you create your own characters?

Brian L. Daniels
537 Pine Ridge Rd
Raleigh, NC 27609

FREE UNIVERSE WITH EVERY TAPE*
AQUAPLANE
COMMODORE 64 SOFTWARE
Ski thru' Marine Menaces, G&Ts, Regattas but beware the Great White Hungry!! AQUAPLANE is an unbelievable original Arcade Game
PURPLE TURTLES
COMMODORE 64 SOFTWARE
A fully animated arcade game with Loveable Turtles, Cuddly Graphics and more Cuteness than you'll find in any other Commodore 64 game! A game for the young at heart and people who have tired of alien bashing
QUINTIC WARRIOR
COMMODORE 64 SOFTWARE
art Man, Part Superman the QUINTIC WARRIOR stands along against the sinister Cratmen and a Domed City gone mad in the distant future. Are you warrior enough to stand by his side in this MEGA- Arcade Game.
RING OF POWER
COMMODORE 64 SOFTWARE
The Colorful King has lost his mind and along with it the Crown Jewels! Now whosoever can find the Jewels shall be proclaimed King. Can you? RING OF POWER is a sophisticated adventure with a GRAPHICS or TEXT option
TIMEX-SINCLAIR 1000, 1500 and 2000 and ACORN ELECTRON.
Send for Advanced Information on our new Timex-Sinclair and Electron Range
SKYHAWK
VIC20 SOFTWARE
Attacking raiders scream out of the sky at you, you check your radar as an explosion blossoms at your side, you return their fire sending one of the attackers plummeting, the fuel gauge flashes a low fuel warning... SKYHAWK runs in 3K or 8K with a Joystick
TORNADO
VIC20 SOFTWARE
Suddenly attacking Colony Fighters leap at me. I dive into their midst firing and still bombing the ground installations below, the sound of explosions rumbles away over the landscape. TORNADO runs on an unexpanded VIC20 + Joystick
QS™
THE GAME LORDS OF BRITAIN
QUICKSILVA INC.
426 West Nakoma
San Antonio, TX 78216
Tel: (512) 340 3684
CIRCLE 34 ON READER SERVICE CARD
Please send me a free color catalog
I currently own or plan to buy a ______ computer
I enclose a stamped, self-addresse envelope
Name
No/Street
City
State
Zip
53

Read This First

Before you enter the programs in this issue, please note:

All the programs require the **8K ROM** and **16K RAM** unless other requirements are given at the top of the first page of the article

Read the article all the way through before trying to enter the program.

A letter after a number shows the type: b for binary; d for decimal; h for hexadecimal.

In PRINT statements:

#: Enter a necessary space.

A (32): The underline means use the graphic on that key. The number in ( ) tells how many times.

A: The overline means use the key in inverse

INPUT: An underlined *word* found on the keyboard should not be spelled out. Enter it directly. If it will not ENTER, hit THEN, then the keyword you want, backspace, delete THEN, and continue entering the line. This memory saving technique may be disregarded if you have enough RAM.

Dear Editor.

It is true that Basic is the most popular and well-supported language for the ZX/TS computers, but, given the power of these mighty little machines, it is evident that much more can be done with them

Specifically, in the language area, implementations with Forth, partial Pascal, and Logo have been produced. A British company recently advertised a subset of Lisp available only in Britain.

These developments certainly whet my appetite, and probably many others are interested in these additions to the range of applications. I am looking for full implementations or subsets of any of the lesser ones, which are usable on the ZX/TS computers, particularly PL/1, APL, Cobol, Fortran, Lisp, "C", Ada, and Prolog.

Peter Mannelli
2034 Grand Concourse
Bronx, NY 10456

Upgrading Problem

Dear Editor.

What are people going to do with all those ZX81s and TS1000s when they move up to larger systems? Someone needs to develop a robot or other toy that can incorporate the computer instead of letting it waste away in a closet.

Glenn A. Osborn
2624 Gunckel
Toledo, OH 43606

How about someone hooking it up with TS2068? Or how about using it to help someone enter the computer age by passing it along to a niece, a nephew, a neighbor, a retired person, the local public library (some are checking them out like books), the local public school (along with **SYNC** *and details about the* **TEC** *Newsletter?)*

Inverse Matrix

Dear Editor

In *SYNC* 3.6 Michael Miller asked for a program that would find the inverse of a matrix. Being an avid Sinclair programmer, I took it as a challenge and wrote a program that calculates and displays the inverse of a 3x3 matrix as well as its determinant. Interested readers may contact me.

Robert H Thompson
Rt. 3, Box 189
Lake City, FL 32055

in and out of sync

Steve Arrants
David Grosjean

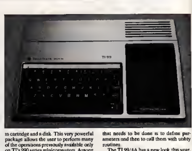
What's a Brand X doing in *SYNC* Magazine?

With improving technology and intensifying competition in the small computer market, more and more computers are available at prices within a few steps of the Timex/Sinclair units. Our sister publication, *Creative Computing*, evaluates many of these systems. We would like to share these reviews with those of you considering another computer.

In addition, we will sometimes take a program or two and show what it would be like to write and run the program on the Brand X computer compared to the Timex/Sinclair. You will probably find these tutorials a useful aid for converting programs from other sources to your Timex/Sinclair computer.

The TI-99/4A

Steve Arrants



Beset with problems from the start, the TI 99/4 was slow to take off. The keyboard was difficult to use, it was too expensive, and it could not be expanded. The only language available was Basic, and software consisted of plug-in modules.

Dropping from a price of $1100 in 1979, to less than $50 today, the TI 99/4A has come the distance. The 99/4A features a 16-bit microprocessor, and color graphics that make other manufacturers drool. A limited typewriter style keyboard is standard, offering upper- and lowercase. Keys may be used in three ways—upper- or lowercase, and as function keys. Keys may be redefined by software, adding to their versatility.

A Peripheral Expension System was made available this year, allowing additional memory and access to disk-based software and versatile printers. A speech synthesizer makes the TI 99/4A one of the few home computers capable of true speech.

The real changes have occurred inside. A new Video Display Processor and a revised operating system aid in the creation of sprites—special graphic characters—and their animation. With the addition of an Extended Basic cartridge, creation of sprites is easily done. All of this work is done by the new VDP Processor—the CPU is left free to work on other information. The CPU initiates sprite action, but the VDP handles everything else.

TI now also provides an editor and assembler module which includes a plug-in cartridge and a disk. This very powerful package allows the user to perform many of the operations previously available only on TI's 990 series minicomputers. Among the features are an Editor which works like a word processor for assembly listings. When so much of the "busy work" has been eliminated, the programmer is free to concentrate on writing better code. The written code is relocatable and linkable. You needn't worry about absolute addresses. Programs can be written in modules, later linked by a loader. Subroutines can be kept on a disk and called into a program when needed.

Users can also access utility routines kept in ROM and GROM. Again, TI has simplified difficult programming tasks. All that needs to be done is to define parameters and then to call them with utility routines.

The TI 99/4A has a new look this year. The familiar black and silver has been replaced by grey, making it appear sleeker. New peripherals are also available. All Hex-bus peripherals designed to work with TI's new CC-40 portable computer will work on the 99/4A. One new peripheral is the Wafertape storage system. Using tiny cassettes, the Wafertape system quickly loads programs into memory. It is different from regular cassettes in that it can search and find a particular program on the tape. At a suggested retail price of $140, it is considerably cheaper than a disk drive and controller. Other new peripherals

Your Timex/Sinclair deserves a Kradle. So do you.

Introductory Price $195.00

We know you'll be happy together.

You're proud of your TS-1000, so why not make its performance even better with the addition of the Kradle. It's got a lot going for it as you'll soon discover. For instance . . .

THE KEYBOARD.

Kradle uses the conventional style keyboard but makes use of snap action switches which give you tactile feedback and a definite "click" for a comfortable touch. A wide, easy-to-find space bar and ten logically chosen extra keys are included. These auto-shifting function keys make the keyboard extremely easy to use.

MEMORY FEATURES.

Kradle expands your computer's memory to 64K bytes of Random Access Memory. And only Kradle lets you copy the computer's ROM into the bottom 8K of memory and, under software control, *turn off the ROM!* This amazing memory circuit also lets you put machine code between 32K and 48K without affecting the video display.

THE GOODIES.

With the cassette enhancer, tapes load easier and more reliably because of Kradle's selective signal filter and squarer.

Kradle was also designed for use with fast cassette software. It suppresses hum and noise while making recorder volume setting less critical.

MORE GOODIES.

- Circuitry. Kradle low power circuits are designed to run cool with your existing power supply.
- Joystick Connector. Allows you to remotely actuate the "Graphic" and four arrow keys with a standard 9-pin TV-game joystick.
- Pilot Light. A power miser LED indicator light lets you know when your computer is powered.
- Expansion Capacity. Kradle lets you add-in economical I/O boards to access the real world. Convenient rear panel cutouts permit ready access for connectors or switches on these boards.

AVAILABLE ADD-INS!

The Kradle Communicator serial card with real RS-232 capability under full software control. The Kradle Manipulator parallel card with 16 (+4) lines of digital I/O (adaptable to parallel printer port.)

K2 ELECTRONICS DESIGN CORPORATION
3000 Varsity Drive • Ann Arbor, MI 48104 • (313) 973-0266

We accept checks, money orders, Visa/MasterCard. Add $2.50 for shipping, additional $2 [illegible] for C.O.D. Michigan residents add 4% sales tax. Personal checks — allow 10 days to clear.

CIRCLE 31 ON READER SERVICE CARD

include a low-cost four-color printer/plotter, and an inexpensive modem.

Milton Bradley, a software supplier to Texas Instruments, has the MBX Expansion System for the 99/4A. This system provides voice recognition capabilities, and ten software packages have been developed for it. Suggested retail price for the MBX Expansion System is $129.95.

Software for the 99/4A has been a problem right from the start. TI never encouraged outside manufacturers to produce software. Indeed, the requirement of a proprietary chip in software cartridges meant that TI had the final word on all software for the 99/4A.

Nevertheless, many third party vendors make software for the 99/4A on cartridge, cassette, and disk. Not as much is available as for the T/S 1000, but the selection is impressive.

Why Buy a Discontinued Machine?

TI has withdrawn from the home computer market, why buy a 99/4A? Because at the closeout prices, the 99/4A is an exceptional bargain. It offers color, sound, a decent keyboard, an exceptionally accurate (but slow) Basic, and a wide range of software for less than an Atari video game.

Why buy a discontinued machine. View ed in this light, why not?

Making a Dancer

David Grosjean

In this issue, we will use the TI-99/4A and the TS1000 to simulate animation in the form of a dancing man. We can easily see some of the strengths and weaknesses of each computer in creating our Fred Astair.

First of all, we must know how we will display the man, because this will determine how we define the parts. For instance, the TS1000 has strong string handling capabilities and strong display commands (e.g., PRINT AT). It has graphics characters that can be entered on the keyboard, but no user definable graphics. Therefore, we can easily put the graphics characters into several strings and PRINT these strings on the screen. In the lines below, 10-40, we define H$ (head), S$ (shoulders), B$ (body), and L$ (legs). S$ and L$ contain four different combinations of the shoulders and legs respectively.

TS1000:

```
10 LET H$= O
20 LET S$=
30 LET B$=
40 LET L$=
```

Graphics notes:

20 8,6,6,5,6,6,6,5,6,6,6,6,8,6,6,6

30 8,5

40. 5,8,space,5,2,4,3,1,8,3,1,2,4

Unfortunately, the TI-99 does not have such strong capabilities or handy commands like PRINT AT, and it has no graphics characters accessible from the keyboard. For our project, its one saving grace is that we can define our own characters. The seven lines below define the head, the body, the curved arms, and the straight legs. We will use some of the predefined characters including the slash, backslash, and the hyphen. To print these characters on the screen, we will uses the CALL HCHAR subprogram which is like PRINT AT except that it can only print characters using the character codes (no strings).

Defining characters is very easy. In the lines below, you will see first, CALL CHAR. This is the command to define a character. The number just inside the parentheses is the character code we will assign it. The letters and numbers in the quotation marks are hexadecimal for the bits of an 8 x 8 character. For a thorough explanation of how this part works, see the manual, it is quite clear on this subject.

TI-99:

```
10 CALL CHAR (43,"18181818FFF
F1818") '
20 CALL CHAR (48,"03030303030
30707")
30 CALL CHAR (91,"COCOCOCOCOC
OEOEO")
40 CALL CHAR (128,"3C42A581A5
99423C")
```

```
50 CALL CHAR (129,"1818181818
181818")
60 CALL CHAR (44,"80C0603818F0
7")
70 CALL CHAR (46,"0103061CF8E
")
```

The next step is randomly choosing the leg and shoulder combinations. On the TS1000, all we do is choose one of four from the leg string and the shoulder string. Line 100 chooses a random number, multiplies it by 4, and adds one to it, giving us one of the numbers 1, 5, 9, and 13, which are the starting points of each of the shoulder combinations Line 110 does the same thing for the leg string

TS1000:

```
100 LET S=4*INT (RND*4)+1
110 LET L=3*INT (RND*4)+1
```

Since we are not using strings on the TI-99, we must choose each character independently. The characters were numbered so that they are one away from another character to be considered for a particular position. For instance, in line 130, RA is the dancer's right arm. There are two possible characters for his arm: our defined one and a hyphen, numbered 44 and 45 respectively. In the parentheses, random number is greater than .5 (true), a -1 will be returned, and if the random number is less than .5 (false), a 0 will be returned. This 0 or -1 will be subtracted from 44, so we get either a 44 will be returned. This 0 or -1 will be subtracted from 44, so we get either a 44 or a 45. A similar process is done for the Left Arm, the Right Leg, and the Left Leg.

TI-99:

```
130 RA=44-(RND>.5)
140 LA=45-(RND>.5)
150 RL=47-(RND>.5)
160 LL=91-(RND>.5)
```

Now we must produce the coordinates to move the dancer. On the TS1000 in lines 50-90 below, P is the horizontal position, and R is a random number. Again we use Boolean operations to decide whether we add or subtract one position to move the man right or left by comparing the random number to .5. (Remember that a true expression on the TS1000 returns a 1, while a true expression on the TI-99 returns a -1.) Lines 80 and 90 check to see if the man is off to either edge and makes the correction to create a complete wraparound instead of having a split body.

TS1000:

```
50 LET R=16
60 LET R=RND
70 LET P=P+(R<.5)-(R>.5)
80 IF R=32 THEN LET P=2
90 IF R<2 THEN LET P=25
```

On the TI-99 version, again life is not so simple. Basically, the same process of creating a wraparound is carried out, except that in an IF-THEN statement, the TI-99 can only GOTO another line number. As a result, we must send the computer to a different line to change P

TI-99:

```
80 P=15
90 R=RND
100 P=P+(R>.5)-(R<.5)
110 IF P=30 THEN 280
120 IF P=2 THEN 300
280 P=3
290 GOTO 130
300 P=29
310 GOTO 130
```

Finally, we move and PRINT the man On the TS1000, lines 120-170 clear the screen, print the man, and return to choose new combinations Note that in lines 130 and 160 we use Boolean operations again. Line 130 chooses a random position for the head. Since we used four characters for the shoulders, the head cannot sit in the middle, so we can make it move back and forth. You can easily see how we choose a shoulder combination—we choose a random number and take the part of the string beginning at that number and including the next three characters The logic in 160 is used because the leg combinations are not of uniform length The open legs (the last combination) is four characters long, while the others are only three. Also, the last two combinations must be printed to the left a space to appear normal (See if you can figure out how the computer uses Boolean operations to determine how much of the string to print and where to print it Keep in mind that L is the starting point in the string of a leg combination.)

TS1000:

```
120 CLS
130 PRINT AT 11,P+(RND>.5),H$
140 PRINT AT 12,P-1,S$(S TO S+3)
150 PRINT AT 13,P,B$
160 PRINT AT 14,P-(L>8),L$(L TO
L+2+(L>5))
170 GOTO 50
```

The TI-99 version is, again, more tricky. We must use the CALL HCHAR subprogram (horizontal character repetition) to print the man, and we must print each part individually In the parentheses, the first number is the vertical position, the second number (P) is the horizontal position, the third number is the character code, and the fourth (optional) is how many times the character is to be repeated. Line 190 clears the screen and line 270 returns to get new combinations

TI-99:

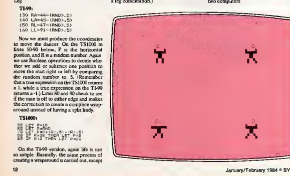
```
190 CALL CLEAR
200 CALL HCHAR(12,P,128)
210 CALL HCHAR(13,P-1,RA)
220 CALL HCHAR(13,P,43)
230 CALL HCHAR(13,P+1,LA)
240 CALL HCHAR(14,P,129)
250 CALL HCHAR(15,P-1,RL)
260 CALL HCHAR(15,P+1,LL)
270 GOTO 90
```

To the TI-99 version, we can add a delay loop with these lines below.

TI-99:

```
170 FOR I=1 TO 50
180 NEXT I
```

If you have followed this program development exactly you should now have a crude dancing man (Remember to use SLOW mode on the TS1000.) You should also see some of the strengths of Boolean operations, as well as the strengths of the two computers ■



sync notes

Paul Grosjean

SYNC in the Home Office

Our theme section "SYNC in the Home Office" offers help for several financial questions most of us have to face sooner or later. In "Building Your Own Spreadsheet" Gordon Young gives an introduction to spreadsheets for record keeping. "Hatching Your Nest Egg" by David Draker lets you consider various options and conditions in evaluating savings plans. "Tax Shelter Time Bomb" from David Lipman finds the point at which a tax saving investment begins to produce taxable income. With "Home Control on a Budget" Bruce Taylor shows how to use your computer to turn devices on and off around the house. Since word processing is part of the modern office, we have a comparative review by Sharon Aker of eight of the packages available. Of course, in using such programs an auxiliary keyboard is very helpful. Lawrence Kelly's roundup and discussion covers most of the options available. Then Sheldon Maloff gives you a break from all this hard work with "Highway Robbery," a fantasy for increasing your capital.

Coming Issues

Our theme section for March/April will be "SYNC at the keyboard." This will focus on programs, programming, and programming tips both in Basic and machine code.

Theme sections under consideration for subsequent issues include a return to "SYNC on the Job," showing how the ZX/TS computers are used in job related situations; "SYNC Goes Shopping," a buyer's guide to products for the TS2068/Spectrum computers, and "SYNC in the Classroom."

Writing for SYNC

SYNC welcomes articles of interest to users of the ZX/TS computers. If you have an article or an idea for one, we would like to see it. Submissions will be returned only if accompanied by a self-addressed envelope with postage attached. If you want acknowledgement of receipt, enclose a self addressed postcard. Programs must be accompanied by a tutorial article discussing at a minimum how the program works. Listings must be submitted on tape unless they are very short. Send a self-addressed stamped envelope for a copy of "Writing for SYNC" which will help you make your manuscript more "editor friendly."

TS1500 and TS2068

By the time you read this the Timex Sinclair 2068 Color Computer and the Timex Sinclair 1500 should be widely available. We have already received enthusiastic reports from various locations around the country.

SYNC will include these new machines in its coverage. We will welcome articles on their use and reviews of hardware and software available for them. Wherever possible we ask that writers include either program listings for both machine types (TS1000/TS2068) or suggestions of where changes may be made.

In *SYNC* 3:4 we ran pictures and a brief description of these new machines. David Ornstein's "Perceptions" column in *SYNC* 3:5 gave a look at the new Basic for the TS2000 series with brief descriptions of what the various commands would do. In *SYNC* 2:4 we published an extended review of the Spectrum from which the TS2068 is adapted.

We just received a production model of the TS2068, and we want to share with you some initial observations and descriptions.

Size and Appearance

Upon opening the box, my first reaction was "It's so big!"—but that reaction comes only if you are used to using the TS1000. The TS2068 measures 17 3/4" x 7 1/2" x 1 3/4". All the plug-ins are made in the rear. An on/off switch is on the left side, and the channel (2 or 3) switch is recessed on the bottom. On the right is the port for Timex's new plug-in software. If you like to work with the computer on your lap, this is just the right size.

Power Supply

The AC power adaptor that comes with the computer is about twice the size of the ZX/TS power packs. The output is 1A at 15VDC.

The Keyboard

All users will find that the keyboard is a significant improvement over the membrane keyboard. The keys are 3/8 x 1/2" chiclet style with spacing slightly wider than the IBM Selectric II keyboard. Although the key action is good, the keyboard obviously will not have the feel of a typewriter keyboard. When a key is pressed, the speaker makes a little noise and provides audio feedback. This makes up partly for the lack of tactile feedback. A full size space bar, shift key on both sides, and the ENTER key at the carriage return position on the IBM typewriters are much more convenient for typing. Touch typists will find this keyboard usable, but still not like a regular typewriter keyboard. The raised dots on the F and J

Microtech Marketing

1418 Lensing Street • Aurora, CO 80010

HANDY ORDER FORM
FILL IN AND MAIL TODAY!

Name ______________________

Address ______________________

City ____________ State ______ Zip ______

Phone # ( ) ______________

2068

MODEL #	QTY	DESCRIPTION	UNIT PRICE	AMOUNT
OUR CATALOG		Refundable off 1st Order	$ 3 00	
TS 2068		Color Computer 72K	179 95	
TS 2020		Computer Cassette Recorder	49 95	
TS 2050		Modem for 2068	109 00	
TS 2040		Printer	89 00	
COMPUTURF #1		TS 1000, 1500 2068, 16K Produces 87% Winners at the Horse Races	29 95	
		TOTAL		
		Colorado State residents please add 7 1% sales tax		
		Handling, Shipping and Insurance	5 50	
		GRAND TOTAL		

2040

2050

CHARGE MY VISA MasterCard

Credit Card No ☐ Visa or ☐ Master Card

Master Card Inter Bank No (above your name) | Expiration Required Mo. Yr

METHOD OF PAYMENT: (check one)

☐ PAYMENT ENCLOSED
in the amount of $ ______ (check or Money Order only)

☐ CHARGE CARD
Information filled in at left

Make checks payable to:
MICROTECH MARKETING
1418 Lensing Street, Aurora, CO 80010

SHIP TO: (Fill in only if address is different from above)

NAME ______________________

ADDRESS ______________________

CITY ____________ STATE ______ ZIP ______

We ship via UPS Please provide Street Address UPS will **NOT** deliver to P.O Box
If you are usually out during the day please leave written instructions for UPS man where to leave parcel

Delivery in 4-6 Weeks

CIRCLE 32 ON READER SERVICE CARD

keys compensate somewhat for the lack of the extra key row on the right of the typewriter keyboard which typists use to help locate the home keys. Touch typists will have to retrain their finger patterns a bit because of the location of some important keys for touch typing such as the period and comma, but this is normal when you switch to another machine

LOADing and SAVEing

One of the biggest problems ZX/TS users have had is in LOADing and SAVEing. The TS2068 is improved in that regard We tried three different tape recorders All loaded at volume settings from 7-10 with the prerecorded tapes from Timex and with tapes of test programs we made, SAVEd, and then LOADed (We should note though that all three recorders have been generally reliable with the ZX/TS computers also, but not in the range found here Also, Timex has released its own tape recorder for computer use for $49 95 Preliminary comments on its capabilities have been very favorable)

The LOADing process is much more convenient It goes through three steps which are shown by color changes on the screen First is the searching pattern The borders alternate between red and cyan When the computer finds the program, it goes into the finding pattern The borders go to a red/blue horizontal strip pattern while the computer is picking up the name of the program The name is displayed If that is the program sought, the LOADing process begins and the loading pattern shows The border shifts to a blue/yellow stripe pattern If the LOAD is unsuccessful, a screen message alerts you. If the program you want is not the first, the computer continues searching and adding program names to the list on the screen until it finds the program instructed In this way you end up with a directory of the tape on the screen

Computer Output

The TS2068 will output to either a TV or a monitor Timex has included a statement pointing out that color TV sets which have total solid state tuning probably cannot be used with the TS2068 and that the AFT control, if any, must be off If your present set cannot be hand tuned, you may have display problems

Since other family members may not want to share color TV time with the computer, this may be a good time to consider adding a dedicated screen A monitor would certainly enable you to take full advantage of the hi res graphics capabilities of the machine

Color

The quality of the color depends greatly on whether you can adjust your TV to get the proper colors for the various keys The hi res color graphics on several tapes from Timex were very well done.

Programming

Since *SYNC* is wholly devoted to helping users of the Sinclair and Timex Sinclair computers get more out of their computers, we will not go into programming at this time That will be the topic of many future articles in *SYNC* However, we will note a few new features that will be of special interest to ZX81 and TS1000 users

1) The FREE command will give you the amount of memory remaining for use. With PRINT FREE, you can find the number of bytes in a line or a whole program This is much more convenient than the byte checkers we have had for the TS1000

2) The MERGE command will do from the keyboard what our "Chaining Programs" article in this issue does. You still must be sure that you have no lines with the same number in the programs to be merged.

3) Both upper and lower case letters are available from the keyboard The commands are printed on the screen in all capitals, but all other letter input will be lower case unless you put the computer in the CAPS LOCK mode See the listings in "Bubbles" and "Where Are You Going?" in this issue

4) All keys have repeat capabilities. No more hitting the delete key once for each character or hitting the right arrow key 32 times to get the cursor to the next line of the program for editing!

5) Inverse video is available directly from the keyboard You can alternate between true and inverse by using these two commands.

6) Twenty locations are available for programming your own characters. The BIT command allows you to arrange all 64 bits of the character matrix into whatever character you want. You assign your character to one of the 20 keys, and then you can use it from that key whenever you want.

Documentation

The users manual for the TS2068 is a significant improvement over the ZX81/TS1000 manuals. It is much more comprehensive and much better illustrated While it is not a complete course in Basic, it will give you a good start in using this computer There is a rush among the book publishers to convert the well-developed Spectrum bookshelf (see *SYNC* 3:5) to the TS2068

TS1000 Convertability

Programs for the TS1000 can be entered on the TS2068, but modifications have to be made since there are some important differences For example, the character sets are numbered differently so character numbers will have to be changed in program transation The display file works differently also See the "Try This" column in this issue.

Spectrum Convertability

Spectrum tapes cannot be LOADed into the TS2068 Many Spectrum software suppliers are busy converting the programs, and we anticipate that by our next issue a number of software packages will be available

Conclusion

The TS2068 will let you take what you have already learned for the ZX81 or TS1000 as a base to build on Some adjustments will have to be made, but you do not have to start from scratch in areas that both types of machine share. Working from this base, you can move on to color, sound, and hi res graphics.

glitchoidz report

The Bookshelf Goes Supernova, 3:5

Contrary to the implication in the article, Tim Hartnell is by no means "retired" from the book field Rather, we understand that he is vigorously engaged in making more titles available to ZX/TS users

ZX Stock Exchange 3:5

1410 32 spaces

The author suggests several improvemants To make the net change in stock prices lower:

```
2310  LET BC=-1
2270: LET BC=1
```

To make the stock split more realistic

```
1995 IF R > 93 THEN LET S(I)=S(I)+.5
```

Try This, 3:6, p. 7, 1st col.

40: Last character is an S

Just for Fun, p. 9.

"Explanation???" is by John Richard Coffey

Just for Fun

"Just for Fun" shares short programs that illustrate a point, demonstrate a technique, or show something the author has found interesting. If you have some programs that you want to share, send them to Just for Fun, SYNC, 39 E. Hanover Ave., Morris Plains, NJ 07950.

If Betsy Ross Had Had a ZX81...

Dan Matejczyk

8K ROM; 2K RAM; 1K RAM

Enter Listing 1. Put your computer in SLOW mode and note how the display develops on the screen. After you are satisfied that you see how the program works, SAVE the program if you want and hit NEW and ENTER.

Now enter Listing 2. Again put your computer in SLOW mode and note the change in how the display unfolds.

These programs show two different ways of producing the same display. The fun in writing them was to develop the display one graphics character or plot statement at a time while minimizing the number of program lines to stay within 1K RAM. Compare the two programs to see how they set up the display.

Dan Matejczyk, 20155 Keswick St., 214, Canoga Park, CA 91306.

Graphics notes
Listing 1:
20 Inverse space
40 A
50, 120 Inverse space
Listing 2:
70. Inverse space
74. A
120 Inverse space

Listing 1.

```
 10 FOR I=1 TO 26
 20 PRINT AT 5,I;"█"
 30 FOR J=6 TO 16 STEP 2
 40 PRINT AT J,I;"▒"
 50 PRINT AT J+1,I;"█"
 60 NEXT J
 70 NEXT I
100 FOR I=1 TO 13
110 FOR J=6 TO 16 STEP 2
120 PRINT AT J,I;"█"
130 NEXT J
140 NEXT I
200 FOR J=23 TO 31 STEP 2
210 FOR I=5 TO 21 STEP 4
220 UNPLOT I,J
230 IF J=23 THEN GOTO 250
240 UNPLOT I+2,J-1
250 NEXT I
260 UNPLOT 25,J
270 NEXT J
```

Listing 2.

```
 53 FOR J=5 TO 17 STEP 2
 60 FOR I=1 TO 26
 70 PRINT AT J,I;"█"
 72 IF J=17 THEN GOTO 80
 74 PRINT AT J+1,I;"▒"
 80 NEXT I
 90 NEXT J
100 FOR J=6 TO 16 STEP 2
110 FOR I=1 TO 13
120 PRINT AT J,I;"█"
130 NEXT I
140 NEXT J
210 FOR J=22 TO 30 STEP 2
220 FOR I=4 TO 24 STEP 4
230 UNPLOT I+1,J+1
232 IF J=22 THEN GOTO 240
233 IF I=24 THEN GOTO 240
234 UNPLOT I+3,J
240 NEXT I
250 NEXT J
```

Think of a Number

Cecil K. Johnson

"Think of a Number" is really a mathematical magic trick. You think of a number, and enter it in the computer. The program reminds you of your selected number and gently suggests an appropriate range of numbers. Of course, you were not told what use would be made of the number in your thoughts so you very possibly thought too high or low. Even though a range of numbers is suggested, no numbers are turned down, and since this is so, the range is quite wide and interesting.

The screen will then display a 4 x 4 magic square made especially for the number in your thoughts. Add rows or columns and you will find that the sum is the secret number that you confided only to the computer. The diagonals will also add up the same, as will the four corner numbers and the four in the center.

You can try another number by pressing RUN and ENTER.

Suppose you thought of 34. Try these combinations: 3 + 5 + 14 + 12, 3 + 10 + 14 + 7, 6 + 10 + 11 + 7, 4 + 15 + 14 + 1. See how many combinations you can find.

Think of some unusual numbers. For example: With 0 the smallest number used is -8, the largest is 9. Both -8 and 8 are used. No number is repeated in a single square. With 70 all numbers are two digit numbers; 10 is the lowest, 25 the highest.

Cecil K. Johnson, 1009 Crest Park Dr., Garland, TX 75042.

Did you notice when you used 70 that the 16 displayed numbers centered around 17? For 34 the numbers center around 9.

Try to find the smallest number that could be used without requiring negative numbers Note that there are no negative numbers used when a magic square for 500 is called up. Could you construct a square for 500 that would use negative numbers?

You might try combining with this Nick Godwin's "Tidying Up Your Display" (*SYNC* 3:1).

Line Notes

30-50 Gently direct the player to an ordinarily acceptable range of numbers

55 Major mathematical calculation

60-95 Locate the numbers on the square

100, 105 Provide spaces to divide the square from the following lines

```
  5 REM "NUMBER" BY CECIL K.
 JOHNSON, GARLAND,TX.  9/8/83"
 10 PRINT "THINK OF A NUMBER."
 20 INPUT A
 25 PRINT "       YOU SELECTED
";A;"."
 30 IF A<15 THEN PRINT "YOUR NU
MBER WILL BE USED BUT IT IS TOO
SMALL."
 35 IF A<31 AND A>14 THEN PRINT
"WE WILL ACCEPT YOUR NUMBER BUT
 WOULD PREFER A LARGER NUMBER."
 40 IF A>30 AND A<65 THEN PRINT
"THAT IS A REASONABLE NUMBER."
 45 IF A>64 AND A<100 THEN PRIN
T "WE WILL ACCEPT YOUR NUMBER BU
T WOULD PREFER A SMALLER NUMBER
."
 50 IF A>99 THEN PRINT "YOUR NU
MBER IS TOO LARGE BUT IT WILL BE
 USED."
 55 LET B=INT ((A-30)/4)
 60 PRINT AT 5,6,B+8,AT 5,12,B+
5,AT 5,18,B+2
 65 IF A=31 THEN PRINT AT 5,24,
12,AT 7,10,16,AT 9,8,14,AT 11,16
,13
 68 IF A=32 THEN PRINT AT 11,10
,14,AT 9,6,15,AT 5,24,17,AT 7,18
,16
 65 IF (A-30)/4=B+.25 THEN PRIN
T AT 9,24,B+10,AT 7,12,B+16,AT 9
,8,B+14,AT 11,18,B+13
 70 PRINT AT 7,6,B+3,AT 7,18,B+
9,AT 7,24,B+4
 75 IF (A-30)/4=B+.5 THEN PRINT
 AT 5,24,B+17,AT 7,12,B+16,AT 9,
8,B+15,AT 11,18,B+14
 80 PRINT AT 9,12,B,AT 9,18,B+7
,AT 9,24,B+10
 85 IF (A-30)/4=B+.75 THEN PRIN
T AT 5,24,B+18,AT 7,12,B+17,AT 9
,8,B+16,AT 11,18,B+15
 90 PRINT AT 11,6,B+6,AT 11,10,
B+11,AT 11,24,B+1
 95 IF (A-30)/4=B THEN PRINT AT
 5,24,B+16,AT 7,12,B+14,AT 9,8,B
+15,AT 11,16,B+12
100 PRINT
105 PRINT
110 PRINT "THIS IS A 4 BY 4 MAG
IC SQUARE"
115 PRINT "FOR ";A;". THE SUM
OF HORIZONTAL"
120 PRINT "LINES OR VERTICAL CO
LUMNS WILL"
125 PRINT "BE YOUR CHOSEN NUMBE
R OF ";A;"."
130 PRINT "ALSO TRY THE DIAGONA
LS AND"
135 PRINT "CORNERS."
```

The Psychic ZX

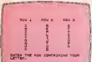
Richard Wilson McDaniel

In a dimly lit room, a stranger said, in a low voice with a hypnotic tone, "Choose a card, for it will lead you to your destiny." I chose and . . .

Actually a friend was showing me a card trick in which he dealt out face up three rows of five cards each He asked me to pick a card and tell him which row the card was in. Then he picked up the rows of cards and dealt them out in the same three row, five card arrangement Again he asked me to tell him which row the card was in Again he picked up the rows, redealt them, and asked which row my card was now in He collected the cards, dealt them face up, and, much to my surprise, found the correct card.

After a few more repetitions of the trick, I convinced my friend to divulge the secret Its application is simple but yet it remains a mystery While waiting in a bus station at 2 30 in the morning, I decided to write a program to duplicate the trick. When I implemented the pro-

Richard Wilson McDaniel PO Box 71, Glasgow, VA 24555



gram, I discovered that various odd numbers of cards in the rows also worked. My version of the trick uses nine cards in each row. The 27 cards are represented by the 26 letters of the alphabet and a space.

I hope this program will mystify and amaze you and your friends as much as it has mystified and amazed me.

Line Notes

10-30: Introduce the program

40-170: Initialize and scramble the letters in R$

180-270: Print the layout of the letters and the prompt for the row containing your letter.

280: Inputs your response

290: Checks the input.

300: Branches to the conclusion of the program after the third question

310-460: Perform the actual shuffling of the letters

470-480: Clear the screen and loop back to the next question.

490: Prints the letter you picked

500-530: Prompt you to try again

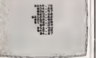
```
10 REM "PSYCHIC"
20 REM RICHARD W. MCDANIEL
30 REM JANUARY 1,1983
40 DIM R$(9,3)
50 FOR A=1 TO 9
60 FOR B=1 TO 3
70 LET R$(A,B)=CHR$(34+A*3+B)
80 NEXT B
90 NEXT A
100 LET R$(9,3)=" "
110 FOR I=1 TO 30
120 LET A=INT (RND*9)+1
130 LET B=INT (RND*3)+1
140 LET T$=R$(A,B)
150 LET R$(A,B)=R$(5,2)
160 LET R$(5,2)=T$
170 NEXT I
180 FOR I=1 TO 3
190 PRINT TAB 4;"ROW 1";TAB 13;
"ROW 2";TAB 22;"ROW 3",,
200 FOR A=1 TO 9
210 PRINT TAB 6;
220 FOR B=1 TO 3
230 PRINT R$(A,B);"########";
240 NEXT B
250 PRINT
260 NEXT A
270 PRINT "ENTER THE ROW CONTAI
NING YOUR   LETTER."
280 INPUT C
290 IF C<1 OR C>3 THEN GOTO 280
300 IF I=3 THEN GOTO 470
310 FOR A=1 TO 9
320 LET T$=R$(A,2)
330 LET R$(A,2)=R$(A,C)
340 LET R$(A,C)=T$
350 NEXT A
360 LET T$=""
370 FOR A=1 TO 3
380 FOR B=1 TO 9
390 LET T$=T$+R$(B,A)
400 NEXT B
410 NEXT A
420 FOR A=1 TO 9
430 FOR B=1 TO 3
440 LET R$(A,B)=T$(A*3+B-3)
450 NEXT B
460 NEXT A
470 CLS
480 NEXT I
490 PRINT ,,"I MYSTICALLY PERCE
IVE YOUR","LETTER TO BE ";R$(5,C
);"."
500 PRINT ,,"WOULD YOU LIKE TO
TEST MY MAGIC ABILITIES AGAIN?(Y
/N)"
510 INPUT T$
520 CLS
530 IF (T$+"#")(1)="Y" THEN GOT
O 180
```

Decimal Alignment

Robert J. Midura

1K RAM

"Decimal Alignment" offers an alternative to Nick Godwin's program for aligning money displays (*SYNC* 3:1). It has only four Basic lines (five, if used as a subroutine). It does not use a FOR-NEXT loop or GOTO branch. Consequently, the routine runs faster and takes less memory. It makes use of the string variables X$ and I$ and the numeric variable N. The stand alone routine occupies 257 bytes.



```
1000 LET I$=STR$ INT N
1005 LET X$=STR$ (INT ((N-INT N)
*100+.01)/100)
1010 PRINT TAB 10-LEN I$,"$",(I$
 AND X$<>"1"),(STR$ (VAL X$+1) A
ND X$="1"),(".00" AND (X$="0" OR
 X$="1")),X$(2 AND X$(1)=".")+(
2 AND (X$(2)="." OR X$="1")) TO
);
1015 IF LEN X$=3 THEN IF X$(2)="
." THEN PRINT "0"
1020 RETURN
```

The routine works only for positive values of N. I have never had the occasion to print negative money amounts. In the financial programs I have written, I put debits in a list and store and print them as positive values. When this list is printed, a heading is required to indicate that the figures are debits.

To use the routine, set the variable N to the number you want printed.

Robert J. Midura, 16 Marmfield St., Worcester, MA 01605.

'You lose!'

Kitchen Sync

Alan Groupe

Shirt-pocket Shaman

> **A pocket reference card should jog your memory, not teach something new.**

A common complaint I have about most computer documentation is that, while the manuals often say too little, the "pocket reference cards" usually say too much. A pocket reference card should not actually teach anything new, but rather just jog your memory of things you have already read in the manual. These cards should contain mostly tables, and rarely, if at all, complete sentences.

A case in point for the need of such a card is the Sinclair ZX81 manual. One of things I often look for is the table of error codes, but I have to flip 22 pages from the back of the book to find it. It is not close at hand.

Consequently, I was very happy to learn that Nanos Systems had added the ZX80/ZX81 to the list of machines for which they produce pocket reference cards.

Nanos Systems Corporation is Shirley A. and Paul P. Nanos of Speedway, Indiana, and has produced much needed reference cards for machines such as the Apple II and TRS-80. When I found that they had done one for the Sinclair, I decided that I had to get one.

The card size (3¾" 8½" folded, 37½" x 8½" unfolded) is a bit more unwieldy than the much smaller "pdp11 Programming Card" that I am used to, but it is still easy enough to work with. Normally, you would only have a couple of panels open at a time.

The card has 20 panels. The 10 panels on one side are chock-full of all sorts of good information about programming the ZX80/ZX81/TS1000 in Basic. The 10 panels on the reverse side are all concerned with programming in Z80 machine language. (I assume that these panels are the same ones found in other Nanos Systems cards for Z80 based machines.)

Panel 1 is taken up entirely by the 22 graphic symbols in the Sinclair character set. It does not seem to me that so much space needs to be devoted to this since the graphic characters are adequately presented in the code chart on panels 5 and 6.

Panels 2 and 3 list the Basic statements, commands, and functions in a very concise and readable format. One section shows the derivation of additional trigonometric functions using the built-in functions of Sinclair 8K Basic. This is a very nice addition to the card, although in all my years I have never really *had* to know how to calculate an inverse hyperbolic cosecant.

The top section of the next panel lists the special characters and operators in Sinclair Basic. The bottom section, entitled "screen layout," puzzled me at first. It turns out to be a chart of the memory displacement and PLOT command Y-coordinates for the 24 PRINT lines on the screen. This chart is correct however, only, if you have the 16K (or more) RAM pack since otherwise the screen is stored compressed.

Panels 5 and 6 show the full character set, in decimal, hexadecimal, and the ZX80 and ZX81 graphics. This is a very handy chart despite the layout (see below).

Panel 7 contains "ZX81—selected ROM calls," though I do not know how the 8 routines listed came to be the chosen ones. Panel 8 contains tutorials on using FOR/NEXT and IF/THEN. As I mentioned in the beginning, I do not believe that tutorials belong on a reference card, but rather in a manual, where they can be more fully detailed. Needless to say, while it certainly did not hurt to have them, I do not feel the usefulness of the card would have been diminished if these two panels had been omitted.

The last two panels describe the ZX80 and the ZX81 memory layout and the Basic error codes. Located at the back, these are very easy to get at.

As good as the front side is, the flip side is not. While the front is very helpful in tracking down Sinclair specifics, I cannot recommend the back side as being particularly helpful in writing machine language programs. The back side is basically divided into two large sections. The first lists all of the Z80 instruction combinations and the length of time to execute each. This is useful if you are trying to optimize code or are building a delay loop.

The other section lists all of the instruction combinations in numerical order (by opcode) along with hexadecimal and decimal counterparts. This is useful if you wish to disassemble some of the monitor.

What is missing, however, is some *convenient* manner of hand assembling from symbolic instructions to a numeric counterpart. A similar list of all instruction combinations in alphabetical order was sorely lacking and would have been very useful.

The layout of the card could have used a bit more thought. For example, the character code chart, which takes up panels 5 and 6, is not placed on facing panels. As a result, you have to open up four panels to use the code chart. This does not matter if you have a large enough table available, but I usually keep the card on my lap.

Still, the Nanos Systems reference card is clear, concise, and, for its shortcomings, is the best handy reference for the ZX80/ZX81 I have seen. If you really like experimenting with your Sinclair, I cannot think of a better way to spend $5.95. It is well worth the price.

From the Panty

One of the things I bought for my new job is a small television set. I figure that if I am going to stay late at work, at least I can watch the network news, and, when it was still on, "Mash." The nicest 5" set that I found was a Panasonic model TR-5091P, which includes an AM/FM radio. This unit cost $124 dollars. It also comes without the radio (for less money I presume)

The reason I am telling you this is not to elicit contributions to pay off my set, but rather to let you in on one of the nicest features of this set.

All portable television sets have a connection for an external antenna. Generally, this connection is a pair of screws for attaching a 300 ohm twin-lead antenna. On this set, however, the connection is a mini-phone jack and it is a 75 ohm input. In other words, this set was designed to be used with portable video recorders, video games, and *home computers!*

After discovering this, I quickly ran out to the local Radio Shack to purchase the appropriate adapter and try this feature out. Radio Shack sells two adapters for this purpose. Number 274-330 is a shielded adapter that costs $1.39 and adapts the Sinclair video cable to the television. Number 42-2444 is a 72" shielded cable that replaces the Sinclair video cable and costs $1.99. Since I am not big on stringing adapters on things, I opted for the cable.

After connecting up the set and fiddling with the brightness and contrast controls a bit (the set is initially adjusted for a TV picture) I got the sharpest and most stable image I have seen from a Sinclair. I also tried it with my ZX80 and found that it, too, produced its best image on that set.

If you are going to use your Sinclair a lot, it makes sense to use a good monitor with it. Not only is this a good monitor, but it is a pretty nice television set, too. The only problem is that now I am stuck buying another one to take to the office.

Build Your Own Spreadsheet

Gordon Young

The term "Spreadsheet" applies to any technique where the user can view portions of a large sheet of records that normally cannot fit onto the screen. Quite often, inventory files require much longer columns than the display can produce and, rather than limit the need, we will simply move our screen like a magnifying glass across 100 columns of information. Likewise, we can scroll upwards or downwards until any 10 of 60 rows is displayed.

The machine language routine we will create is, above all, fast! With a minimum of keystrokes, the user can find any record in the file within seconds. Here, our records are to be stored into A$ only, and our headings for the columns in T$. This header allows us to distinguish each column of data in A$ at all times. Though the title string remains on the screen, it does move left or right with the records below. A$ is DIMensioned to 60,100 and occupies 6K of memory. We can make our file larger, but this is tricky and an explanation will be given later in this discussion. T$ is a single string, DIMensioned to 100 and its contents are defined by the user for his own program.

Arrow keys 5 to 8 move the screen records about the sheet to reveal information in the entire file. It would be nice to have 20 records printed onto the display, but, as you will see, the moving records can be distracting. So we will stick with 10 (you can easily alter this later). Since we have 4 directions of travel, we need to highlight the first printed record by inverting its video characters so that it is more distinguished than the others.

As the machine code loop is always busy finding and printing records to the screen, the only way to return to Basic is by pressing M for a return to the menu or C for a correction during the display period.

Figure 1. REM statement.

```
REM 0...............................
1...............................
2...............................
3...............................
4...............................
5...............................
6...............................
7...............................
8...............................
9...............................
0...............................
1...............
```

Less than 400 bytes are used to do all the work of producing a spreadsheet, and only simple requirements are mandatory:

A$=All File Records	A$=DIMensioned (60,100)
T$=Title String (heading)	T$=DIMensioned (100)

The user can alter portions of the program to:

1. Print up to 15 records (controlled by addresses 16731 and 16657)
2. Slow down/Speed Up movement of screen (delay address=16640)
3. Alter file size (described later)
4. Change routine escape keys from "M" or "C" to user's choice.
5. Highlight any particular record in the display other than the first.

Begin by putting the computer in the FAST mode (the reason for this will become obvious). Enter line 1 REM (followed by 395 periods). As the computer runs your program and comes upon the word REM, all contents afterwards are not operated upon until it approaches the next line number. This is where our machine code is stored. Later, when we want to call upon the routine, we will force the machine to operate on it directly. After line 1 is entered, it should look like Figure 1 (each segment of 32 characters has been intentionally numbered for count reference here). SAVE what you have so far on tape, and return the computer to SLOW.

Figure 2. MC loader.

```
9900 LET N=16514
9905 LET M=0
9910 INPUT A
9915 IF A>300 THEN GOTO 9960
9920 POKE N,A
9925 PRINT AT 20,M;A
9930 LET N=N+1
9935 LET M=M+5
9940 IF M<28 THEN GOTO 9910
9945 LET M=0
9950 SCROLL
9955 GOTO 9910
9960 IF A>555 THEN GOTO 9999
9965 IF A=555 THEN GOTO 9990
9970 LET N=N-1
9975 IF M<20 THEN LET M=M-5
9980 PRINT AT 20,M
9985 GOTO 9910
9990 SAVE "ASSEMBLER"
9995 GOTO 9910
9999 STOP
```

At this point, you are ready to POKE the machine code listing data into the REM line by typing in the Basic assembler in Figure 2. Again, SAVE to tape. You may want to use an assembler of this type for any machine language program that begins at address 16514 (the first byte of data following a REM statement in line 1). If you fear that you may accidentally erase line 1, enter: POKE 16510,0.

This assembler will accept your machine language codes and automatically insert them into the REM line. With it, you can make corrections to wrong entries (enter 333), SAVE your data at any time during entry (enter 555) or stop the assembly process totally (enter 999). All codes in Figure 3 must be read from left to right, top to bottom. RUN the assembler and begin entering the listing codes. Occasionally SAVE your work by entering 555 (this is a number that cannot be POKED into any address). The screen display will correspond to the listing and aid in data reference. It prints 6 entries across the SCROLLs. Enter 999 to stop the assembler when finished.

If you have doubts about the 300+ codes you just entered, a simple checksum routine will verify your data. Eliminate the assembler and type in the checksum:

```
9995 LET M=0
```

Gordon Young, 4616 N. River Rd. #27, Oceanside, CA 92056

50 easy new games, graphics and business programs for your Timex Sinclair 1000

Now you can start getting more fun, more challenge and more practical value from your computer with the clearest, most comprehensive book ever written for the TS 1000, compatible for use with Sinclair ZX-81.

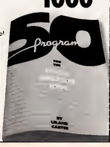
In *50 Programs for the Timex Sinclair 1000*, Leland B. Carter outlines exciting new ways to use your unexpanded Timex for games, graphics, math, physics, business, puzzles and more.

You'll find hours of sheer enjoyment. You'll instantly expand your software library. What's more, you'll permanently enhance your computer programming skills.

Step-by-step instructions in Sinclair Basic let you immediately type in and run programs like:

- Booby Trap
- 2-D Reversal
- Target Gallery
- Flight Graphic
- Stopwatch
- Calendar
- Calculator
- Blast-Um
- ESP
- What's the Pattern

And many, many more!



Each program has been thoroughly tested and re-tested for ease of entry. And you'll find the user-friendly statements extremely helpful in guiding your progress, and in suggesting further input.

Plus you'll enjoy better-quality graphic and screen display than you've ever achieved on your TS 1000.

50 Programs for the Timex Sinclair 1000 is truly the best software investment you can make right now. At just $6.95, you're getting each program for less than 14 cents. Compare that to the price of pre-packaged programs!

Start getting more from your computer by ordering your copy of *50 Programs for the Timex Sinclair 1000* today!

Compatible for use with Sinclair ZX81

For faster service, PHONE TOLL FREE
800-631-8112
(In NJ only 201 540-0445)

MAIL TODAY TO
CREATIVE COMPUTING PRESS
Dept. NJ9F 39 East Hanover Avenue
Morris Plains NJ 07950

Please send me_______ copies of *50 Programs for Timex Sinclair 1000* for only $6.95 each, plus $1.50 shipping & handling per book

Total Amount $_______

☐ Payment Enclosed (CA, NJ and NY State residents please add applicable sales tax.)

Name_______________
(please print)

Address_______________

City_______________

State__________ Zip__________

*Outside U.S., add $3 for shipping and handling.

☐ Check here to receive a FREE catalog of computing books, magazines and guides

Also available in your local bookstore or computer store

Figure 3 MC listing

```
0 0 14 35 42 42 126 24 254 64 64 40 24 84 40 34 1 130 192 205 [illegible] 12 35 [illegible] 40 50 254 64 134 58 14 32 50 254 64 [illegible] 62 16 3 254 237 58 4 139 100 136 17 0 130 14 20 24 81 2 19 50 64 58 7 [illegible]
0 0 0 81 16 3 254 230 0 55 1 5 34 42 [illegible] 132 166 64 2 189 50 11 137 58 134 34 254 64 135 86 [illegible] 135 0 254 64 10 64 35 103 35 135 35 64 0 64 121 40 64 33 32 1 35 24 61 136 254 137 254 64
0 0 33 254 64 35 117 62 32 134 100 237 132 132 35 64 0 205 58 [illegible] 137 128 64 134 50 64 [illegible] 0 61 64 137 68 64 40 33 60 50 128 24 40 35 64 61 42 237 14 61 2 237 58 25 19 231 254 64 0 64 40 201
0 0 130 0 126 24 40 7 250 64 0 74 64 84 81 42 237 167 0 5 64 177 254 64 134 86 14 40 60 254 64 135 58 14 35 88 136 254 248 3 35 254 24 12 74 33 35 24 91 138 28 118 254 35 0 81 194 254 40 [illegible]
0 59 64 32 254 248 3 35 34 42 254 81 56 254 24 18 74 2 68 0 32 35 254 [illegible] [illegible] 58 7 0 254 64 135 56 [illegible] 135 64 21 35 35 35 0 248 64 23 26 19 243 132 54 152 13 0 58 35 58 [illegible] 58 [illegible] 27
0 0 113 249 198 35 35 81 132 132 0 64 135 0 248 64 34 124 [illegible] 0 251 32 50 69 64 [illegible] [illegible] 81 [illegible] 13 [illegible] 64 42 40 126 24 35 40 74 [illegible] [illegible] 254 [illegible] 64 254 128 121 40 65 250 136 65 40 195 [illegible]
```

```
9995 FOR N=16514 TO 16870
9997 LET M=M+PEEK N
9998 NEXT N
9999 PRINT M
```

Enter RUN 9995 You should get a sum (the sum of all entries in the machine code program) of 29192. If not, you need to PEEK all addresses from 16514 to 16870 and check your data.

Our example program, a real estate listing, will need 100 columns with a heading (T$) for TYPE/PRICE/1st MORTGAGE/RATE/LOAN (type)/ADDRESS/CITY/AGENCY, and PHONE number of agent. Figure 4 shows the initialization beginning at line 10 with the header string at line 30 T$ must be DIMensioned before A$. If not, it will incorporate an unnecessary delay in the file display movement.

Enter and RUN the Basic sample program Insert a few fictitious records to fill the record file and return to the MENU. Now VIEW the file by pressing "2" in the MENU. The screen will almost instantly display 10 file records Pressing the arrow keys will move the spreadsheet in the corresponding directions Press M to return to the MENU

Figure 4 Basic listing.

```
  5 LET A=1
 10 DIM T$(100)
 20 DIM A$(A0 100
 30 LET T$=" [illegible]
 40 FOR N=1 TO 80
 50 FAST
 60 LET R$(N)="           $           $
 70 LET R$(N,2 TO 3)=STR$ N
 80 NEXT N
 90 SLOW
105 CLS
110 PRINT AT 5 11 "[illegible]" AT 7,8 "(ENTER CHOICE)" AT 10 3 "1. ENTER A RECORD." AT 12 5, "2. VIEW SPREADSHEET."
120 IF INKEY$="1" THEN GOTO 200
130 IF INKEY$="2" THEN GOTO 150
140 GOTO 100
150 CLS
160 PRINT "REAL ESTATE SPREADSHEET" AT 16,0 "[illegible]" TAB 6 "PRESS M FOR MENU."
170 RAND USR 16624
180 GOTO 100
200 CLS
210 PRINT "ENTER RECORD INFORMATION."
220 PRINT "1. TYPE? (4 LETTERS)"
230 INPUT R$(R 1 TO 4)
240 PRINT "2. PRICE?"
250 INPUT X
260 LET R$(R 13-LEN STR$ X TO 12)=STR$ X
270 PRINT "3. FIRST MORTGAGE PRICE?"
280 INPUT X
290 LET R$(R,21-LEN STR$ X TO 20)=STR$ X
300 PRINT "4. 1ST MTG. LOAN RATE?"
310 INPUT X
320 LET R$(R 31-LEN STR$ X TO 30)=STR$ X
330 PRINT "5. TYPE OF LOAN (VA-FHA-CONV.)"
340 INPUT R$(R,32 TO 35)
350 PRINT "6. PROPERTY ADDRESS?"
360 INPUT R$(R 37 TO 67)
370 PRINT "7. CITY?"
380 INPUT R$(R,69 TO 83)
390 PRINT "8. AGENCY OFFERING PROPERTY?"
400 INPUT R$(R 85 TO 91)
410 PRINT "9. AGENCIES PHONE NUMBER"
420 INPUT R$(R 93 TO 100)
430 LET R=R+1
440 GOTO 100
```

To make corrections during the display period. "C" from the spreadsheet activity This gives you the option of stopping the routine where it stands and making a correction. A correction at a column to the far right on the sheet requires you to reenter the routine without readjusting the display to position (column) #1 The reentry point is address 16607. Your program might go something like this (sample program)

```
180 IF PEEK 16521=50
    THEN GOTO 100
182 IF PEEK 16521=40
    THEN GOTO 185
184 GOTO 100
185 PRINT AT 15,4, ''ENTER
     CORRECTED ADDRESS''
188 INPUT A$ (PEEK 16518+1,37
    TO 67)
190 RAND USR 16607
192 GOTO 180
```

Adding 1 to the contents of address 16518 will give the number of the first record in the display Address 16521 holds the character code of the key M (50) or C (40) The machine code is activated again in line 190

If the spreadsheet moves too fast, you can increase the delay 16640 presently has 12, but 24 will double the delay and can be POKEd until the movement is comfortable Likewise, smaller numbers will increase the speed If you need to use letters other than M or C to escape the routine, addresses 16860 and 16865 should be POKEd with the code of the appropriate characters (refer to the back of the user manual)

To change the number of records displayed is a little more involved Address 16731 should be POKEd with the number of records you want printed (1 to 17) This means you have to POKE 16657 with the number of records in the file minus the number printed on the screen To print 17 records then you need to POKE 16731,17 and POKE 16657, (60-17) This will disturb the highlighted record, and address 16814 should be POKEd with the number of records being printed (POKE 16814,17 for this example) This is also the address that affects the highlighted record Normally, this would be 10 if 10 are being displayed To move the highlighted record to the second one being displayed would require you to POKE 16814,9

Now, if you need to store more information with larger files, some big changes must be made! Some alternative solutions along with the address changes are found in Figure 5

Figure 5.

A$ DIMensioned	(50,115)	(80,100)	(100,100)
T$ DIMensioned	(115)	(100)	(100)
POKE address 16552	123	69	21
POKE address 16576	115	100	100
POKE address 16657	40	70	90
POKE address 16713	83	68	68
POKE address 16749	118	103	103
POKE address 16837	83	68	68
File Size	5 75K	8K	10K
Approx Memory Avail	10 25K	8K	6K

Tax Shelter Time Bomb

David Lipman

Practically all tax shelters have a built-in time bomb Most of those that do not are real tax shelters with a real loss of capital

Let me describe a seemingly perfect tax shelter I own a swampy piece of land that was under water for at least two months each year I valued the land at $1,000 and would have sold it for less.

John, a friendly site locator for a major gasoline company, suggested that I build a gasoline station to their specifications and lease it to them The offer was rather generous they would pay $100 per month as land rental, arrange zoning changes, get all permits, contract to fill the swamp, supervise building construction and arrange for a mortgage The mortgage would be large enough to pay for everything including a $20,000 "fee" for legal service and construction supervision. The total rental was set at exactly the monthly mortgage payment plus the land rent plus a variable amount equal to the real estate taxes The gasoline company was to be responsible for all maintenance and insurance, including my liability insurance.

It was a perfect situation I received an annual return equal to 120% of my asking price for the land, the gasoline company real estate department showed an income of $20,000, and I had a tax deduction of about $2,500 for the first year Thirty years in the future I would own a gasoline station with a potential monthly rental in excess of the land value.

Wonderful! In thirty years I would have a fully funded generous retirement plan plus all those nice tax deductions along the way Remember all this took place in 1953 when $1,000 per month was a pretty decent income and there were no IRA or KEOUGH plans available at the neighborhood bank

David Lipman, 9 West Castle Ave. Spring Valley NY 10977

But wait a minute, what happens when the monthly principal payment on the mortgage exceeds the depreciation? At that point a taxable income is generated with no funds available to pay the tax. When does this happen? I wrote

```
INPUT STARTING DATE OF MORGAGE
(FORMAT=MO YEAR)   1.1983

INPUT TERM OF MTGE  MONTHS  360

INPUT ANNUAL INTEREST RATE  12

INPUT AMOUNT OF MTG  $50200

INPUT ANNUAL STRAIGHT LINE DEPRE
CIATION  $2000

A SMALL TAXABLE INCOME WILL
OCCUR IN THE 2 MONTH OF 2003
112 MONTHS BEFORE END OF MORTGAG
E.
TAXABLE INCOME WILL INCREASE
UNTIL IT REACHES $3758.55 IN
THE LAST YEAR OF THE MORTGAGE
```

down the method of calculating a mortgage amortization (algorithm, fore-runner to a program) P-(R-iP)=P' What I was looking for was the month in which (R-iP)=Dep, so I labelled the columns on my accountants pad P, ΔP My LAX Monroe Calculator (mechanical) was called into service, and I sat in front of it for a half hour totally mesmerized while the discs whirled and clattered I wrote the numbers in the P and ΔP columns As I dozed off, I realized that what I was struggling to find was really ΔP the first differential of the compound interest equation

$$P = R \quad \frac{(1+i)^{n}-1}{i(1+i)^{n}}$$

I put away the Monroe LAX, picked up my K & E Slide Rule, set the first differential equal to depreciation and 30 seconds later had the answer 17 years, 11 months from inception of the mortgage, a small taxable income would be generated The amount of taxable income would increase until the last year of the mortgage when it would be about $5,100 00

Armed with this knowledge I was able to plan properly so I was not hurt by the mathematics and income tax

The problem can be solved for *n* by a brute force trial and error mortgage table comparison process which is long and unwieldy (90 seconds) It can be shortened by using this program which will run in a little over one second

The formulas used on the Sinclair ZX81 are.

$$N = \frac{[10/(1-0.6B^{2})]+8\frac{B}{9}}{1-\frac{i^{2}}{2}}$$

$$\text{Where } B = \frac{(\frac{R}{D}[\frac{1}{1+i^{2}}]) - 1}{(\frac{R}{D}[\frac{1}{1+i^{2}}]) + 1}$$

D = Monthly Depreciation
R = Monthly Payment
i = Monthly Interest, Decimal
N = Number of Months

To bring the story up to date, last week my friendly site locator telephoned from Florida where he lives in semi-retirement "David," he said excitedly, "Do I have a tax shelter for you! There's this shopping center and apartment house available . "

Line Notes:

20-30 Input instruction, starting date of mortgage,

40-60 Error trap for incorrect month and year

70-230 Input instructions, prints

240-280 Calculation of monthly mortgage payment

290-340 Calculation of first taxable income, months before end of mortgage.

350-380 Calculation of taxable income, last year of mortgage.

NEW LOW PRICE

BYTE-BACK modulas

64-K MEMORY

With **Battery Backup** **$109.95 KIT**

$119.95 ASSEMBLED & TESTED

the "ULTIMATE MEMQRY"
UM-64

FEATURES:

- **Battery Backup - Computar and full 94-K Backed**
- **PROM/ROM spckat**
- **Raaat Switch-Raaat without dastroying Program**
- **BYTE-BACK EXCLUSIVE FULL 64-K**

The 0-8K area is available You can execute a copy routine (provided) to copy the TIMEX ROM into the 0-8K area of RAM then flip a switch and you have your operating system in RAM You can modify it and create your own customized operating system Full details examples & programs included

WHY PAY MORE FOR LESS FEATURES?
GET THE "ULTIMATE MEMORY"
BYTE BACK'S UM-64

ALL NEW

MD-2 SMART MODEM

In Stock!

$119.95 KIT

ASSEMBLED & TESTED $149 95

With New **SMART** , Menu Driven SDFTWARE included FREE

- Sand and racelva Programs by Phone
- Copy Information into memory from a Host Computar (such as **Compusarve**), Print it, Raview it, Save it on Tapa, Sand Text from Memory
- Usa Timex 2040 Printar or Any RS-232 Printer
- RS-232 Printar Port providad
- No axtra memory Required, But with 64-K Mamory You Can Store Up To 60 Full Screens

COMPUSERVE PACKAGE WITH 5 FREE HOURS ONLY $39 95

NEW ZX PRO/FILE

Cassette & Book $16.95 including shipping

Clearly the Best File Management Program Available
This New Program is what the Timex has been waiting for

At **BYTE-BACK** we have used & evaluated almost every data organizer & Data File Program on the market & we were not really impressed until we tryed **Thomas Woods NEW ZX PRO/FILE** Program It is truly the best piece of software we have seen for the Timex Computer We now use it to help run our business This program is so FAST that even when the program is full the life you want is on the screen within a second

The ZX PRO/FILE has features not found on the other Data File programs It has **FILE ANALYSIS** function which allows you to count analyze frequency %, etc. Capability to create **FILES OF ANY SIZE** in the same program **MULTIPLE WORD SEARCH**, excellent **ADD/EDIT** features **ORDERED FILE OUTPUT** based on any numerical value contained in the files. **DEFINABLE PRINTER FUNCTIONS & AUTO SEARCH** Works with both **16K & 64K** memory

FREE 59 page **BOOK.** This book alone is worth the purchase price ZX PRO FILE's machine language concepts & methods are fully explained It includes a complete **PROGRAM LISTING &** detailed explanation of how the program works & how it can be easily modified It also includes a machine language programming section

BYTE-BACK'S BB-1
CONTROL MODULE
$59.00 KIT *In Stock!*

ASSEMBLED & TESTED $69 00

- **8 Independent Relays** *(with LED status indicators)*
- **8 Independent TTL Inputs** *(with Schmitt trigger buffers)*

- By using a single POKE command you can change and latch the status of each of the 8 relays
- Your computer can read the status of all 8 inputs by the use of a single PEEK command
- A comprehensive manual is included that has complete application details
- More than one BB-1 can be used at a time

16 K MEMORY UM-16 $59.95 KIT $69 95 assembled & tested

Battery backup, reset switch, PROM/ROM aockat PLUS...
1 year, 100% trade-in credit towerds the UM-64

RS-232 Module $59.95

ASSEMBLED & TESTED $69 95

Drives the Gorilla/Bannana & Othar Serial Printers

ALL MODULES CARRY 90-DAY WARRANTY
TRY BYTE BACK MODULES FOR 10 DAYS WITH NO OBLIGATION

NEW

Gorilla/Banana Printer

Prints on Slandard Papar 80 Columns
Specify Sarial or Parrallal

Discounted to only **$239.95**

See "Leading Edge" Ads

BYTE-BACK KEYBOARD ~~$89.95~~ **$59.95**

We use tha same key switch found in the axpensive computers such as Hewlett Packard Taxas Instruments, Atari DEC atc The multi-colored key caps ara the same type used on the IBM Parsonal Computer, but with tha TIMEX lagens

EXPERIENCE QUALITY

Shipping and Handling $4 95 EXCEPT ZX PRO/FILE

OROER PHONE (803) 532-5812

ITEMS ORDERED: ______

Bill My ☐ Am Exp ☐ Visa ☐ M/C
Chacks Accepted

Exp Data ______ Card No ______
Name ______
Addresa ______
City/Stata/Zip ______
Phone ______

Dealer Discounts Available

Mail To **BYTE-BACK CO.**
Rt 3, Box 147, Brodie Road
Leesville S.C 29070

CIRCLE 10 ON READER SERVICE CARD

390-470. Conversion of months before end of mortgage to month, year

480-550 Print result of calculation.

Use RUN to initiate program. To exit use STOP. The running time is 2 seconds.

An answer indicating taxable income before beginning date of mortgage means that there is no tax shelter.

Error statement A/320 means that no taxable income occurs during term of the mortgage.

List of Variables:

- Y: Beginning date, (MO. YEAR) of mortgage
- T: Term of mortgage, months
- I: Percent interest, annual
- P: Principal amount of original mortgage
- D: Depreciation, annual
- A: Intermediate variable, partial solution compound interest mortgage payment equation.
- R: Monthly mortgage payment
- C,B,E,F,G: Intermediate partial solutions
- N: Solution number of months to end of mortgage
- Q: Intermediate variable, partial solutions compound interest mortgage equation, last year
- L: Principal payment less annual depreciation for last year of mortgage
- H: Years until "N" (partial year shown as decimal)
- M_1: Conversion of decimal year (H) to month
- M_2: Sum of beginning month and ending month
- M_3: Correction if M2 is greater than 12 (December)
- Z_1: Beginning year
- Z_2: Sum of beginning year and years until "N"
- Z_3: Correction if M2 is greater than 12 (December)

An expanded program that calculates all of the IRS acceptable depreciation schedules is available on tape for $9.95 from the author requires 16K RAM.

```
10 REM "DRP" PROGRAM TO CALCU
LATE WHEN TAXABLE INCOME OCCURS.
D. LIPMAN 5/30/83
20 PRINT "INPUT STARTING DATE
OF MORGAGE (FORMAT=MO.YEAR)"
30 INPUT Y
40 IF Y<1 OR Y>12 THEN GOTO 20
50 LET YY=(Y-INT Y)*10000
60 IF YY<1900 OR YY>2100 THEN
GOTO 20
70 PRINT Y
80 PRINT
90 PRINT "INPUT TERM OF MTGE
MONTHS"
100 INPUT T
110 PRINT T
120 PRINT
130 PRINT "INPUT ANNUAL INTERE
ST RATE"
140 INPUT I
150 PRINT I
160 PRINT
170 PRINT "INPUT AMOUNT OF MTG
: $"
180 INPUT P
190 PRINT P
200 PRINT
210 PRINT "INPUT ANNUAL STRAIG
HT LINE DEPRECIATION $"
220 INPUT D
230 PRINT D
240 FAST
250 LET I=I/1200
260 LET D=D/12
270 LET A=(1+I)**T
280 LET R=P*I/(1-(1/A))
290 LET C=R/D
300 LET E=I/(I+I*I)
310 LET B=(C*E-1)/(C*E+1)
320 LET F=(10/(1-0.8*B**2)+6)*B
/G
330 LET G=I-(I**2/2)
340 LET N=F/G
350 LET Q=1/(1+I)**12
360 LET L=R*(1-Q)/I
370 LET L=INT (L*100)/100
380 LET L=L-D*12
390 LET H=(T-N)/12
400 LET Z=(Y-INT Y)*10000
410 LET Z=Z+INT H
420 LET M=(H-INT H)*12
430 LET M=M+INT Y+1
440 IF M>=13 THEN GOTO 460
450 GOTO 480
460 LET Z=Z+1
470 LET M=M-12
480 PRINT
490 SLOW
500 PRINT "A SMALL TAXABLE I
NCOME WILL OCCUR"
510 PRINT "IN THE ";INT M;" M
ONTH OF ";Z;
520 PRINT INT N;" MONTHS BEFORE
END OF MORTGAGE."
530 PRINT "TAXABLE INCOME WILL
INCREASE"
540 PRINT "UNTIL IT REACHES $
";L;" IN THE"
550 PRINT "LAST YEAR OF THE MO
RTGAGE"
```

Home Control on a Budget

Bruce C. Taylor

You can operate virtually any electrical device in your house with your ZX81 or TS1000 computer, an inexpensive wireless home controller, and an adaptation of the circuits used in my "Robotics on a Budget" article (*SYNC* 3:4).

The control system adapted for use in this project is the BSR X-10, available through several sources including Heath and Sears. The control modules which are connected to lamps, appliances, or wall switches, receive a signal transmitted over the house power wiring and thus require no direct hookup to the computer. The controller is available for around $35 and the modules for about $17.

Modifying the BSR X-10

To modify the BSR X-10 for this project, first remove the screw from the deep hole in the center of the bottom of the X-10 case. The two halves of the case will then pull apart. Now remove the circuit board from the top half, with the 28 pin IC on it, by first removing the five screws.

Be careful to keep the keyboard upside down (IC upright) because, when you remove the circuit board, there will be nothing to keep the keypad buttons from falling out.

After the screws are removed, carefully pry the circuit board out and turn it over to expose the foil circuit side of the board (Photo 1). The total IC pinout to the keypad switches is listed in Figure 1. Prepare eleven hookup wires each about 6" long. Solder one end of each to an IC pin circuit foil (see Figure 2 and Photo 1). After the wires are soldered, push them flat against the board and bend them over the edge of the board as shown in the photo. Then push the board back into the top half of the case. There will just be enough room for the wires between the case side and the board edge (Photo 2) if all wires are lined up and none overlap each other.

Bruce C. Taylor, 9765 E. Sierra, Tucson, AZ 85748.

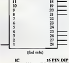
Figure 1.

BSR X-10 Control IC Switch Pinout.

Switch	Control	IC Pins
1	1	17
2	17	28
3	1	20
4	20	28
5	1	18
6	18	28
7	1	19
8	19	28
9	1	18
10	16	28
11	1	27
12	27	28
13	1	21
14	21	28
15	1	26
16	26	28
ON	20	25
OFF	25	27
BRIGHT	19	25
DIM	25	26
ALL LIGHTS ON	18	25
ALL OFF	21	25

Photo 1

Figure 2. BSR X-16 Switch Control IC.



IC CONNECTION	to	16 PIN DIP JUMPER SOCKET
1		1
16		2
17		3
18		4
19		5
20		6
21		7
25		8
26		9
27		10
28		11

Photo 2

The RX81 board drives a Darlington transistor array, activating 5V 14 pin DIP relays which switch the BXR Z-10.

Be careful thet you do not dump out the keypad switch buttons Keep the top half of the case upside down until the board is all the way back into the case top When you see that it fits (you have formed the wires against the edge of the board), remove the circuit board again.

Next, cut a small rectangular hole in the side of the case or drill some holes so that a 16 pin DIP socket can be attached later to the outside of the case with the hookup wires from the IC soldered to it. The jumper pinout is listed in Figure 3 Now pull the wires through the hole in the case as you replace the circuit board into the top half of the case for the final time (Photo 2)

Next, cut the hookup wires about an inch long outside the hole in the case and solder them to the DIP socket (Photo 3 and Figure 3) Then glue or otherwise fasten the DIP socket to the outside of the case

Photo 3

Figure 3

Summary of Switch Connections to 16 Pin Jumper.

SWITCH	JUMPER PINOUT		RELAY #
1	1	3	1
2	3	11	8
3	1	6	9
4	6	11	10
5	1	4	11
6	4	11	12
7	1	5	13
8	5	11	14
ON	6	8	2
OFF	8	10	3
BRIGHT	5	8	4
DIM	8	9	5
ALL LIGHTS ON	4	8	6
ALL OFF	7	8	7

Pinout for remaining switches not used in this project

9	1	2
10	2	11
11	1	10
12	10	11
13	1	7
14	7	11
15	1	9
16	9	11

You have now completed the modification to the BSR X-10 After the rest of the project is finished, you will be ahle to plug a DIP jumper cable into the socket on the X-10 (Photo 4)

The Boards

For the interface between the BSR X-10 and the computer I used the Zodex RX81 input/output boards in combination with the Computer Continuum Buffered Bus Development Board The RX81 outputs (D0-6) will be connected to transistor switch and rely circuits Other output control circuits for the ZX/TS computers could also be easily adapted to this design, e.g , Byte-Back's BB-1 control module with built-in relays

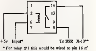
The Zodex RX81 board (also available assembled as a "Control Board for 8 devices") can be plugged directly into the ZX/TS bus If you want to add more boards, a simple Y connector will do the trick.

Photo 4

The Computer Continuum expansion board was chosen because its 3 amp capacity for 5V supply allows for additional circuits However, you must know which of the two versions of the board you have, neither of which can be used for this project without some modification.

The earlier version of the board will accept the Zodex board plugged directly into an expansion edge connector (50 pin, 1 inch centers) soldered to the CC board, but the logic will not work without an additional simple decoder circuit

Figure 4. Dip Relay



* For relay #1 this would be wired to pin 16 of trans array #1

** For relay #1 these would be wired to pins 1 and 3 of jumper socket.

Darlington Transistor Array

Photo 5a.

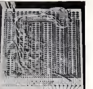


Photo 5b.

WANT TO BUILD YOUR TIMEX/SINCLAIR INTO A REAL COMPUTER?

STATEMENT: The BASICARE MICROSYSTEM is a group of modules which allow you to begin with the least expensive computer available (TS1000) and build a powerful, flexible SYSTEM with exactly the capabilities you need. THIS IS NOT A SIMPLE MEMORY ADDON! By using the PERSONA module (see below) you provide an interface which permits use of additional memory (to 1 Megabyte or more), ROM capsules, printers, a variety of input/output configurations, defineable character capability, A/D or D/A and 16 bit processor (8086). In addition, it will soon be possible to add modules for RS232, floppy disk or high speed tape. Each module measures 7.3"x3.4"x.8". The modules stack on one another. Sinclair Users Network calls it "The ultimate in hardware expansion systems".

A HEAVY DUTY POWER SUPPLY SHOULD BE USED WITH 3 OR MORE MODULES

PERSONA — Connects to the rear of the computer with a flexible cable and acts as the **interface for the other modules**. This module, and the others which stack on it, is exceptionally attractive. Neither drawings or photos do the system justice. It buffers all the computer's signals via the ribbon cable. It then talks to the other modules through a 64 pin ORGANIC BUS. Other modules respond when PERSONA sends their individual SLOT signals along the bus.
020-000 N $69.95

MINIMAP — A mapping device which can expand address space to as much as 1 megabyte. When used with RAM 64 (from 1 to 16 RAM 64s) space is organized into vertical 64K pages. These pages are divided into SEGMENTS labeled ROM, TOOL, FILE A, DATA, PATH, SLOT, FILE B.

It is possible to have many BASIC programs, and several TOOL and DATA SEGMENTS, distributed among different PAGEs. The SEGMENTS may be "slid" relative to one another so that BASIC programs can use TOOL and DATA from a different page or so many BASIC programs can be simultaneously resident in different PAGEs.

It is possible to instantly switch between these different programs. The output of one program can also be used as the input of another program. RAM 08 and DROM can be used in TOOL and DATA positions and TOOLKIT in a TOOL SEGMENT. If you have a RAM 16 it could be used for FILE A while RAM 64 could fill four pages of FILE A or B.
.... $59.95

RAM 16 — Sits on top of PERSONA to add 16K of RAM **$54.95**

RAM 64 — 64K arranged as four blocks of 16K **all of which can be used simultaneously** under the control of MINIMAP. On board address decoding allows simultaneous use of many RAMs in conjunction with the MINIMAP **$149.95**

RAM 08 — A low cost memory expandable from 2K to 8K. Located in an address space not used by BASIC, either 2000 to 4000 or 8000 to A000. Ideal for data or machine coded routines shared by several programs. More useful than you might expect **$39.95**

2K Ram plug in chips for Ram 08 .... $11.95

TOOLKIT — Can use up to four 2K EPROMs. Address space is located after the on-board ROM. This allows the calling of TOOLKIT held routines using the USR function. Several TOOLKIT modules can be used under MINIMAP control as they are individually enabled/disabled.
020-007 N $44.95

LINK a — 8 channel A to D **$54.95**

DROM — Memory storage with its own rechargable battery to provide non-volatile memory. Expandable from 2K to 8K. Individual 2K blocks can be protected against accidental overwriting
.... **$89.95**

2R add one for DROM $13.95

USERFONT — User definable characters for use with DROM and TOOLKIT. User can define a set of 128 display characters dynamically under program control, when used with DROM or RAM 08. Also available in TOOLKIT where the alternative characters are stored in an EPROM. USERFONT is ordered in conjunction with the module it will be used with.
020-012 H $17.95

PERICON a — 24 lines of Programmable I/O. 3-8 bit ports. Uses Intel 8255 A. Up to 4 PERICON a can be used **$64.95**

PERICON b — 24 buffered lines of heavy duty OUTPUT. Each can sink 40mA from 15V supply. 4 can be used. **$59.95**

PERICON c — Centronics printer interface **$74.95**

CENTRONICS CABLE $19.95

INVERSE VIDEO MODULE
Solders inside case
$16.00

REPEAT KEY MODULE
Solders inside case
$15.00

2X AD — Assembler/Debugger
$20.00

2XIS - Disassembler **$20.00**

POWER SUPPLY — 9V with ON/OFF switch. More powerful and cooler running than the standard power supply **$23.95**

SWITCHED OUTLET BOX — One outlet is offset and switched for your power pack. The other three outlets are left live. No more plugging and unplugging or turning off everything when you want to turn the computer on and off **$10.95**

16 KRAM — (Audio Comp.) **$15.00**

We realize many Timex/Sinclair owners hesitate to spend the money required to buy an 80 column printer or a disk drive. However, our 80 column printers and disk drives can be removed and used on other computers by simply buying the proper interface when needed. You do not have to worry about wasting money when you buy our systems.

COMPUSA FLOPPY DISK SYSTEMS

COMPUSA FDC-101 — Can handle 40 track single or double density disks. Up to 250K bytes. SAVE and LOAD with no PEEK, POKE or USR. Does not interfer with tape save and load. Includes interface, power supply, one Shugart SA 200 disk drive and cabinet capable of holding 2 disk drives. Just plug in and use. List $499.00
.... **Sinclair Place $460.00**
Shipping $8.00

FDC-102 — Same as FDC 101 but with 2 disk drives. List $760.00
.... **Sinclair Place $740.00**
Shipping $10.00

VDTEM — A/D interface measures voltage and temperature. Probe provided for liquid or air temperature. Resolution better than .0005V or 0.04 °C. 1V voltage range can be expanded. Pulls less than 20MA at +11V unregulated **$59.95**

THURNAL ELECTRONICS

Modular system. All units are cased and plug together. No wiring, soldering or modifications needed. Can be used with RAM packs, printers, etc. These are very well built units with wide application for control or education.

I/O PORT — 4½"Wx3½"Lx1½"H. Fits between computer and RAM pack (if used). Used to connect controls, sound and music generators, printers, floppy discs, light pens, data transmission, etc. 16 fully programmable input/output lines which are divided into two 8 line ports. Supports full handshaking for data transfer, interrupt facilities and subtle bi-directional & control modes. All outputs T T L compatible **$34.00**

MOTHERBOARD — 3¼"Wx6¼"Lx⅞"H. Plugs on the I/O port. Used for multiple add-ons. Can handle up to 16 lines **$30.00**

4 CHANNEL RELAY BOX — 3½"Wx2½"Lx1½"H. Plugs into output port of I/O or motherboard. Allows programmable switching of loads to 1.5A, 240VAC or 3A; 110VAC/24"DC. 4 single pole contact relays. **$27.00**

TRANSISTOR DRIVERS* **$18.00**

8 WAY SWITCH UNIT **$24.00**

8 WAY INDICATOR UNIT* **$24.00**

8 WAY A/D CONVERTER **$46.75**

**Requires 9V power source*

Z EXCHEQUER — The Klutz-Proof Banking Program. Whether you are an expert or just getting started this program will work for you. This is the best documented, easy to use banking program you'll find. All you need to know is how to load a tape into your machine and follow SIMPLE instructions. Do deposits, checks, charges, debits, automatic teller transactions, fees and so on. Has a complete reconciliation routine. There are 28 categories for classifying checks and deposits. These are easily changed. If desired, Simple PRINT command for hard copy. Complete program listing included **$10.00**

MEMOTECH SPECIALS

	LIST	OUR
Memopak RAM-32K	$ 79.95	$ 73.00
64K	$139.95	$120.30
Memopak Hi Res Graphics	$ 79.95	$ 73.00
Memopak Assembler	$ 39.95	$ 38.00
Memopak Centronics Interface	$ 59.95	$ 57.00
Memopak RS232 Interface	$ 79.95	$ 73.00
RS232 or Centronics Cable		$ 19.95
GP100A Printer Package	$299.00	$280.00
Includes interface and cable (same as Gorilla Banana)		*shipping $10.00*
GP250X Printer Package	$379.00	$368.00
Includes interface and cable		*shipping $10.00*
Memocalc	$ 39.95	$ 38.00
Memotext	$ 39.95	$ 38.00
Memotech Keyboard	$ 79.95	$ 73.00
Mura Modem	$149.94	$140.00

Includes RS232 cable and compuserve

We will replace any defective products. All payments in U.S. dollars. Add $2.00 for 4th class or $3.50 for 1st class shipping or to Hawaii, Alaska or Canada (except printers). VISA & MASTERCARD add 3%.
Washington residents add 7.8% tax.

Sinclair Place • P.O. BOX 2288 • REDMOND, WA 98052

you will have to build. The Sinclair printer will not work without this decoder circuit either.

The newer version of the CC expansion board comes with the decoder circuit built into the board, but the expansion pad pinout has been reversed so that the Zodex board can no longer be plugged directly onto the CC board. The Zodex board can be plugged onto the bus connection for the 16K RAM, but then you have to work out another location for the RAM.

The easiest way to tell the two versions of the CC board apart is that the newer version has a 74LS27 IC chip located next to the optional LM323 voltage regulator while the older version does not.

If you have the older version, you can plug the Zodex board directly onto an edge connector on the C board. Although there are several ways to install the required decoder circuit, I recommend writing to Computer Continuum for a copy of the documentation for the new version. Figure 1, schematic, and Figure 2, legend, are all you will need. Then install the 74LS27 just as it is on the new version. It can be installed in the same spot as the new version with an IC socket, a few jumper wires, and some cuts in the circuit board foil. This will give you essentially the newer version.

If you have the new version of the Computer Continuum board, the decoder is already installed. If you do not want to solder the Zodex board(s) directly together, you will have to build an intermediate connector. The pinouts of the .1 inch center edge connectors are jumpered to the correct fingers of the .156 inch center, 4 x 5 inch PC board.

The circuit I will describe uses the RX81 board to drive a Darlington transistor array, activating 5 volt 14 pin DIP relays which in turn switch the BSR X-10. The transistor array and DIP relay pinouts are shown in Figure 4.

As with the robot circuit I plugged the RX81 boards into the Computer Continuum expansion board, and will not repeat the description of that hookup here.

The only additional PC board for this project is pictured in Photo 5. Again I used the OK Hobby Board and plugged it into the Computer Continuum board with electrical interconnection only for +5 volts and ground. The photo shows a partial wire-up with the transistor array at the upper right (the socket below it is for a second transistor array); five adjacent DIP relay sockets (relays are installed only in four); and, the 16 pin DIP socket at the upper left is the output to the X-10. The flat cable in the upper right corner is a jumper from the RX81 output. A complete pinout for the project is listed in Figure 5.

Walking through One Circuit

For an example of the end-to-end wiring of the project, I will walk you through one control circuit. RX81 output D1 (referred to as 1-1) is connected to pin #2 of transistor array #1. The corresponding output of the array is pin #15. This is connected to pin #6 of relay #2. Pin #8 of relay #2 is connected to pin #6 of the DIP jumper socket. In order to complete the switch action of the relay contacts, pin #14 of the same relay is connected to pin #8 of the DIP jumper socket. This line controls the "ON" switch of the X-10.

Photo 6

Remember that ground and +5 volt connections also have to be made to each transistor array and relay (Figure 4). The +5 volts is connected to pin 2 of each relay and pin 9 of each transistor array. Also, a ground connection is made to pin 8 of the transistor array. Note that the ground connection to the coil of each relay is made by the output of the transistor array.

After all wiring is complete, the relay board is connected to the BSR X-10 with a 16 pin double male ended DIP jumper cable (see Photo 6). The RX81 boards and Computer Continuum board are also visable in the photo. My entire computer setup with BSR X-10 sitting on top of the case housing the expansion board is pictured in Photo 7.

Figure 5.

RX-81 Output	Darlington Transistor Arrays*			DIP Relays			Controls Switch
D0-6	Trans Array #	Input Pin #	Output Pin #	Pin 8 Input to Pin 6 of Relay #	Pin 14 to DIP Jumper Socket Pin #	to DIP Jumper Socket Pin #	
1-1	1	1	16	1	1	3	1
1-2	1	2	15	2	6	8	ON
1-3	1	3	14	3	8	10	OFF
1-4	1	4	13	4	5	8	BRIGHT
1-5	1	5	12	5	8	9	DIM
1-6	1	6	11	6	4	8	ALL LIGHTS ON
1-7	1	7	10	7	7	8	ALL OFF
2-1	2	1	16	8	3	11	2
2-2	2	2	15	9	1	6	3
2-3	2	3	14	10	6	11	4
2-4	2	4	13	11	1	4	5
2-5	2	5	12	12	4	11	6
2-6	2	6	11	13	1	5	7
2-7	2	7	10	14	5	11	8

RX-81 output board #1 wired as "out 7"
RX 81 output board #2 wired as "out 6"

Example: Output D0 from board wired as "out 7" is listed above as 1-1.

*ULN2003A

Figure 6. Software Commands.

Function/Switch	Relay	Output	POKE A,	POKE B,	Keyboard Entry
1	1	1-1	1	7	1
ON	2	1-2	2	7	0
OFF	3	1-3	4	7	F
BRIGHT	4	1-4	8	7	B
DIM	5	1-5	16	7	D
ALL LIGHTS ON	6	1-6	32	7	L
ALL OFF	7	1-7	64	7	A
2	8	2-1	1	6	2
3	9	2-2	2	6	3
4	10	2-3	4	6	4
5	11	2-4	8	6	5
6	12	2-5	16	6	6
7	13	2-6	32	6	7
8	14	2-7	64	6	8

(Note: A=16522 and B=16524)

CIRCLE 38 ON READER SERVICE CARD

Photo 7

Start Controlling

Now that the system is wired up, it is time to punch up some programs and start controlling. The data sheet that comes with the RX81 explains a few programming instructions to get you started, but we need to show off the controlling capabilities with a sample program such as in Listing 1 and Listing 2.

Both programs start with the line of machine language in the REM statement followed by POKE 16517, 79

To activate an output line you first POKE the binary number for the line, then POKE the number of the output board (if you have more than one board) and the activate the command with an OUT USR statement. For example, if you wanted to activate output line #3 on an output board wired as "OUT 7" you would write the program as follows

```
100 POKE 16522,4
(addresses line #3)
110 POKE 16524,7
(addresses board wired as OUT 7)
120 LET OUT=USR 16521
(activates command)
```

Figure 6 summarizes the software commands. In this application you POKE a command which latches a switch on, pause a short time to activate the X-10 function/switch to a key on the ZX/TS keyboard as listed in Figure 6. To turn on appliance or lamp module "1" you first press 1 on the keyboard and then 0. To turn all lamp modules on press A.

Of course, you can work out other control programs. You are limited only by your imagination.

Parts List

BSR X-10 Control System
  Heathkit #GDP-1510
Lamp Module
  Heathkit #GDP-1512
Appliance Module
  Heathkit #GDP-1514
ULN2003 Transistor Darlington Array
  Mouser #511-L203B
Magnacraft 5 colt DIP Relays
  Knapco #171DIP262
16 Pin Single Ended DIP Jumper
  Digi-key #R112-6
16 Pin Double Ended Dip Jumper
  Digi-key #R116-18
PC Board. Hobby Board.
Digi-key #K160-ND
Edgeboard connectors.
  Digi-key #C1-22
  Digi-key #C5-50
Computer Continuum expansion board
Zodex RX-81

"Aha!"


TIMEWORKS

Listing 1.

```
 2 REM <=J TAN Y PEEK TAN
10 POKE 16517 70
20 LET A=16522
30 LET B=16524
40 LET C=16621
```

Addresses:

Heathkit, Heath Co., Benton Harbor, MI 49022

Mouser Electronics, 11433 Woodside Ave., Santee, CA 92071

Knapp of Florida, Inc., 4750 96th St. N., St. Petersburg, FL 33708

Digi-key Corp., PO Box 677, Thief River Falls, MN 56701

Computer Continuum, 301-16 Ave., San Francisco, CA 94118

Zodex, East Hill, Oakham, MA 01068

```
 50 POKE A,1
 60 POKE B,6
 70 LET OUT=USR C
 80 STOP
 90 GOSUB 530
100 STOP
110 POKE A,2
120 POKE B,6
130 LET OUT=USR C
140 STOP
150 GOSUB 530
160 STOP
170 POKE A,4
180 POKE B,6
190 LET OUT=USR C
200 STOP
210 GOSUB 530
220 STOP
230 POKE A,8
240 POKE B,6
250 LET OUT=USR C
260 STOP
270 GOSUB 550
280 STOP
290 POKE A,16
300 POKE B,7
310 LET OUT=USR C
320 STOP
330 GOSUB 570
340 STOP
350 POKE A,32
360 POKE B,7
370 LET OUT=USR C
380 STOP
390 GOSUB 570
400 STOP
410 POKE A,64
420 POKE B,7
430 LET OUT=USR C
440 STOP
450 GOSUB 570
460 STOP
470 POKE A,128
480 POKE B,7
490 LET OUT=USR C
500 STOP
510 GOSUB 570
520 STOP
530 POKE A,0
540 POKE B,6
550 LET OUT=USP C
560 RETURN
570 POKE A,0
580 POKE B,7
590 LET OUT=USG C
600 RETURN
```

Listing 2.

```
  2 REM <=J TAN Y PEEK TAN
 10 POKE 16517 70
 20 GOTO 2060
 30 POKE A,0
 40 POKE B,7
 50 LET OUT=USR C
 60 PAUSE 30
 70 POKE A,10
 80 GOSUB E
 90 PAUSE 70
100 GOSUB D
110 IF RND<.5 THEN GOTO 140
120 POKE A,5
130 GOTO 135
140 POKE A,9
150 GOSUB E
160 PAUSE 37
170 GOSUB D
180 GOTO 2210
190 POKE A,0
200 LET OUT=USR C
210 PAUSE 30
220 RETURN
230 LET OUT=USR C
240 RETURN
250 POKE A,0
260 POKE B,7
270 LET OUT=USR C
280 PAUSE 30
290 POKE A,10
300 GOSUB E
310 PAUSE 37
320 GOSUB D
330 POKE A,9
340 GOSUB E
350 PAUSE 5
360 GOSUB D
370 GOTO 2210
380 POKE A,0
390 POKE B,7
400 LET OUT=USR C
410 PAUSE 30
420 POKE A,10
430 GOSUB E
440 PAUSE 37
450 GOSUB D
460 POKE A,6
470 GOSUB E
480 PAUSE 5
490 GOSUB D
500 GOTO 2210
510 STOP
520 REN ARNODN SELECT
530 IF RND<.67 THEN GOTO 2250
540 POKE A,0
550 POKE B,7
560 GOSUB E
570 POKE A,10
580 GOSUB E
590 PAUSE 100
600 POKE A,0
610 GOSUB E
620 POKE A,6
630 GOSUB E
640 PAUSE 37
650 POKE A,0
660 GOSUB E
670 POKE A,6
680 GOSUB D
690 FOR U=1 TO 12
700 LET IN=USR F
710 IF IN=4 THEN GOTO 790
720 IF IN=1 THEN GOTO 790
730 IF IN=5 THEN GOTO 750
740 IF U=12 THEN GOTO 790
750 NEXT U
760 GOSUB D
770 GOTO 380
780 GOSUB D
790 POKE A,10
800 GOSUB G
810 FOR U=1 TO 60
820 IF U=20 THEN GOTO 840
830 NEXT U
840 GOSUB D
850 POKE A,6
860 GOSUB E
870 PAUSE 30
880 POKE A,0
890 GOSUB E
900 POKE A,5
910 GOSUB G
920 FOR U=1 TO 10
930 LET IN=USR F
940 IF IN=2 THEN GOTO 960
950 IF IN=1 THEN GOTO 990
960 IF IN=3 THEN GOTO 680
970 IF U=12 THEN GOTO 1010
980 NEXT U
990 GOSUB D
1000 GOTO 250
1010 GOSUB D
1020 POKE A,10
1030 GOSUB G
1040 FOR U=1 TO 10
1050 IF U=10 THEN GOTO 1070
1060 NEXT U
1070 GOSUB D
1080 POKE A,9
1090 GOSUB E
1100 PAUSE 37
1110 POKE A,0
1120 POKE B,7
1130 GOSUB E
1140 POKE A,32
1150 POKE B,7
1160 GOSUB G
1170 PAUSE 60
1180 GOSUB D
1190 POKE A,16
1200 GOSUB E
1210 PAUSE 100
1220 GOSUB D
1230 POKE A,32
1240 GOSUB E
1250 PAUSE 45
1260 GOSUB D
1270 POKE A,144
1280 GOSUB E
1290 POKE A,16
1300 GOSUB G
1310 PAUSE 30
1320 POKE A,164
1330 GOSUB E
1340 PAUSE 20
1350 GOSUB D
1360 POKE A,128
1370 POKE B,7
1380 GOSUB E
1390 PAUSE 70
1400 GOSUB D
1410 FOR U=1 TO 300
1420 LET IN=USR F
1430 IF IN=128 THEN GOTO 1460
1440 IF U=300 THEN GOTO 1460
1450 NEXT U
1460 POKE A,64
1470 POKE B,7
1480 GOSUB E
1490 POKE A,68
1500 POKE B,6
1510 GOSUB E
1520 PAUSE 60
1530 POKE A,33
1540 GOSUB G
1550 PAUSE 10
1560 POKE A,97
1570 GOSUB E
1580 PAUSE 5
1590 GOSUB D
1600 POKE A,70
1610 POKE B,7
1620 GOSUB G
1630 PAUSE 220
1640 POKE A,64
1650 GOSUB E
1660 POKE A,99
1670 GOSUB G
1680 PAUSE 200
1690 POKE A,64
1700 GOSUB E
1710 POKE A,144
1720 POKE B,6
1730 GOSUB E
1740 POKE A,18
1750 GOSUB G
1760 PAUSE 10
1770 POKE A,148
1780 GOSUB E
1790 PAUSE 20
1800 GOSUB D
1810 LET IN=USR F
1820 IF IN=0 THEN GOTO 1840
1830 PAUSE 400
1840 POKE A,128
1850 POKE B,7
1860 GOSUB E
1870 POKE A,96
1880 POKE B,6
1890 GOSUB E
1900 PAUSE 100
1910 POKE A,33
1920 GOSUB G
1930 PAUSE 10
1940 POKE A,101
1950 GOSUB E
1960 GOSUB D
1970 POKE A,64
1980 POKE B,7
1990 GOSUB E
2000 PAUSE 40
2010 POKE A,6
2020 GOSUB G
2030 PAUSE 230
2040 GOSUB D
2050 GOTO 2210
2060 LET A=16520
2070 LET B=16524
2080 LET C=16521
2090 LET D=190
2100 LET E=200
2110 LET F=16514
2120 LET G=230
2130 POKE A,0
2140 POKE B,7
2150 LET OUT=USR C
2160 POKE A,0
2170 POKE B,6
2180 LET OUT=USR C
2190 PAUSE 400
2200 LET RN=1
2210 POKE A,5
2220 POKE B,7
2230 LET OUT=USR C
2240 LET TN=0
2250 LET TN=TN+1
2260 LET IN=USR F
2270 IF IN=1 THEN GOTO 30
2280 IF IN=2 THEN GOTO 630
2290 IF IN=4 THEN GOTO 380
2300 IF IN=3 THEN GOTO 250
2310 IF IN=6 THEN GOTO 380
2320 IF IN=7 THEN GOTO 30
2330 IF TN=20 THEN GOTO 530
2340 GOTO 2250
```

try this

"Try This" features short programs to show off your computer, impress your family and friends, and tickle your imagination when *SYNC* arrives at your place. Send your contributions to: Try This, SYNC, 39 E Hanover Ave., Morris Plains, NJ 07950

Type in the following lines.

TS1000:

```
10 SCROLL
20 IF INKEY$<    THEN PRINT TA
B CODE INKEY$; "
30 RUN
```

Put the computer in SLOW mode. Press RUN and ENTER. Then press various keys. Can you control the site of the launch pad? Our thanks to

Hung Ton
6837 Carnegie Dr
Richmond, VA 23226

Type in the following lines.

TS1000:

```
10 LET L=INT (RND*21)+1
11 LET L1=INT (RND*19 +1
12 LET J=INT (RND*255 +1
20 FOR F=0 TO L
30 PRINT AT L1,F CHR$ J
40 NEXT F
50 PRINT AT L1,L CHR$ J
60 FOR F=(L-1) TO 0 STEP -1
70 PRINT AT L1,F,
80 NEXT F
90 RUN
```

TS2068: Make these changes

```
12 LET J=INT  RND*223)+33
90 GO TO 10
```

Press RUN and ENTER. Observe the results. Our thanks to

Robert Hewson
12397 Lansdowne
Detroit, MI 48224

Type in the following lines:

TS1000:

```
20 LET L=1
30 LET L=L+150
40 DIM U$(L)
70 PRINT AT 10 15
72 PRINT AT 10 15
80 IF L 15E3 THEN PRINT AT 15
   0  OVER 15K
90 GOTO 30
```

TS2068: Make these changes

```
70 PRINT AT 10 15 "x"  PAUSE 7
80 IF L>35E3 THEN PRINT AT 15
0  over 35K
```

Press RUN and ENTER. Observe the progressive slowing down. This demonstrates the use of a time/memory comparison. Our thanks to

Robert A. Manion
1231 Peermont Ave
Pittsburgh, PA 15216

Hatch Your Nest Egg

David Draker

The best part about money is spending it The worst part is that eventually the money runs out. Calculating how long the money will last can be time consuming and tedious. How frustrating it is to complete a lengthy set of calculations covering a 20 year period, only to discover that you forgot to take into account a car purchase in year 14, or that you made a mistake in year 3, or that you now wish you had used a different interest rate in your calculations.

A colleague of mine who will retire soon wanted to know what kind of income he could anticipate during his retirement years He asked me if I had a formula to calculate how long the money would last if he received 7% interest on his severance pay and withdrew $300 a month.

I did not have a simple formula, but I thought immediately of my Sinclair and decided to write a program to let the computer do the work

Program Features

Estimating future income and expenditures is difficult because of the large number of factors that must be considered such as initial capital amount, new additions of capital, interest rates, interest accumulation, and monthly withdrawals

Initial Capital

You begin using the program by entering a lump sum amount of capital, e.g., your existing savings, a bond coming due, you equity from the sale of a property, your annual savings from a raise in salary, an estimate of your inheritance, or simply your dream of your lottery winnings Any amount of initial capital may be used

David Draker, 92 Ruskin St., Ottawa, Ont., Canada K1Y 4B2

The display shows a summary of your inputs and prints the years and the amount of capital left until it is all gone.

New Capital

You may then choose: 1) to add new capital in lump sum amounts during any 5 years, 2) not to add new capital, or 3) to add capital in only 1 or 2 years, up to a maximum of 5 years. You can select any year, e.g number 4, number 13, and number 17

Interest Rates

The program then offers you a choice of holding the interest rate constant or of varying the interest rate. The rate may be your own estimate, or be based on information from newspapers, trade magazines, your banker, or an investment advisory service.

1) Constant Interest

If you feel that no one can predict interest rates or that, with inflation moderating, the interest rates will not be as volatile as in the last decade, you may choose the present interest rate (or a more moderate one) and hold that rate constant for all the years of the calculations

2) Varying Interest

If you have faith in professionals' ability to predict financial matters or feel that interest rate volatility will continue, you will probably choose to vary the interest rate assumptions. If you choose the varying interest rate option, the program recognizes that near term predictions will be more accurate than longer term predictions This recognition requires changing the number of years for which a given interest rate will apply.

Since a fairly good estimate of interest rates may be possible for the first year, the program asks you to specify one interest rate for the first year Predictions then become difficult The program asks you to specify an interest rate for the next 2 years, and a third for the following 3 years. To this point, interest rates have been selected for the first 6 years. Accurate prediction beyond that point is virtually impossible. Thus, the program asks you to select a fourth interest rate for all remaining years of the calculations, probably a moderate rate reflecting normal economic times.

Interest Accumulation

The next choice in the program is whether or not to leave the capital alone for a number of years to collect interest before you begin withdrawing money For example, if you have an annuity or a savings plan for your children's education, you will probably not withdraw money for a certain number of years. The capital amount will then grow at the rate you previously determined, according to your selection of interest rates and the additions of new capital

On the other hand, if you intend to begin withdrawing money immediately from your capital, you would not select this option.

Withdrawal Rates

The final choice of the program is either a constant or a varying monthly withdrawal rate.

If you are relatively certain about your intended withdrawals and of the stability of the economic environment, then you would probably choose a constant withdrawal rate.

If you believe that inflation will continue, or if you estimate that your withdrawal needs will change over the years (for example, because of your living expenses going up or down), then you would select the option to vary the withdrawal rates.

The program assumes that your prediction of your financial needs will probably be more accurate in the near term than in the long term. Thus, you are asked to select one monthly withdrawal rate for the first 2 years, another for the next 3 years, a third for the following 5 years, and a fourth for all remaining years.

Regardless of whether you select a constant or varying withdrawal rate, the withdrawals are not introduced into the calculations until after the years of interest accumulation without withdrawals have ended.

Program Output

At this point, your inputs are finished. The program will first display a summary of your selected options and inputs and then print the years and the amount of capital left until all the capital is gone.

Thus, you will see how many years the money will last under your set of assumptions. If the results are not satisfactory in terms of meeting your financial objectives, you can change the items and try the calculations again.

Examples

The following three examples demonstrate the flexibility of "Nest Egg." I recommend that you use the FAST mode when running this program.

1) Retirement

Upon retirement, many people receive a lump sum payment of severance benefits. Suppose that you retire at age 63 and receive $10,000. Because of a pension, you will not need to withdraw money immediately, so you decide to leave the capital alone for 4 years to collect interest. At age 71, you decide to sell your house and move to an apartment. Thus, you will have both an influx of capital and an increase in monthly living expenditures.

LOAD or ENTER the program in Listing 1. Press RUN and ENTER to start. First, the program will ask you to enter the initial capital. Enter 10,000. When the computer asks if you wish to add new capital, answer yes, and, upon prompting for the number of such years, enter 1. The computer asks you to enter the year number and the amount of new capital. Since at the age of 71 you would be at year number 8 of the program calculations (which began at your age of 63), enter 8 and then enter the capital amount (e.g., 60,000 for the house sale).

Then select interest rates. If you believe rates will change over the years, answer yes to the computer's questioning if you wish to vary the rates. The computer will then ask for the interest rate for the first year (enter 8), the next 2 years (7), the next 3 years (6.5), and all remaining years (4).

You are then asked if you wish to leave the capital alone to collect interest for a number of years. In this case, you answer yes, and upon prompting for the number of years, enter 4.

Now you are asked to decide if your withdrawals will vary over the years. Since you are changing accommodations, answer yes. Perhaps you also wish to increase the rate of withdrawals to cover inflation. Remember that you retired at age 63 and let interest accumulate for 4 years until age 67. At age 71, you will see an increase in expenses as you move to an apartment.

After estimating your monthly financial needs, enter upon prompts a withdrawal rate of 150 per month for the first 2 years (i.e., to age 69), 200 per month in the next 3 years (i.e., to age 72), 600 per month in the next 5 years (i.e., to age 77), and 700 per month in all remaining years. The dramatic increase in the third withdrawal rate is to take into account the increased accommodation expenses.

As soon as you enter the last figure, the program prepares and displays the following summary of your inputs.

At this point, you can review the inputs to see if you wish to change any assumption when you run the program again.

Press CONT and ENTER, and the program will begin the calculations. When the display reappears after about 10 seconds, it will show 20 years of calculations. Note that the amount of capital increases to $13167 at the end of year 4, reflecting the 4 years of interest accumulation you selected. The capital left then decreases to $9264 at the end of year 7 because the withdrawals began in year 5. In year 8, the capital increases to $69634 as the house sale was included at the beginning of the year. The next year, the capital increases again to $70019, because interest income was slightly larger than withdrawals. But the next year, the capital decreases as the withdrawal rate changes from $200 monthly to $600 monthly (i.e., year 10 is year number 6 of monthly withdrawals). For the five years of $600 monthly withdrawals, the capital declines significantly from $70019 in year 9 to $46188 in year 14. From that point, monthly withdrawals of $700 produce a more rapid decline to $2722 in year 20. Press CONT and ENTER.

If your child, as a university student, withdraws $250 a month, how long . . . ?

When the display reappears, it shows the last year of calculation, year 21, with capital left of $5570. This last figure means that after 21 years, you would be $5570 in debt, or that your Nest Egg would be gone in between 20 and 21 years. Thus, with this set of assumptions, you would exhaust your money at about age 83 or 84.

Given a realistic life expectancy, that situation is probably acceptable. If it does not seem quite enough, then you may wish to rerun the calculations, changing the amounts of initial and new capital, and the years of interest accumulation without withdrawals.

If you wish to test new assumptions, simply enter yes when asked by the program if you wish to do so.

2) Saving Plan for Children's Education

Say upon the birth of a child, you put $500 aside as the start of an edueational savings plan. Say you add an additional $500 per year for the next 5 years, and then let the money collect interest until the child is 18. Suppose the child, as a university student, will need to withdraw the money at a rate of $250 per month to help cover educational costs (presumably in addition to other sources of income.)

Upon prompting, enter 500 as the initial capital, and yes, there will be years of new capital being introduced. Enter 5 to answer how many years, and, upon prompting, enter 2 as the year and 500 as the amount and so on, up to and including year 6. If you wish to make a conservative estimate of interest rates, answer no, you do not wish to vary them. "Nest Egg" will then ask you what constant interest rate you wish to use for all years. Say you enter 5. Enter yes when asked if the capital will be left alone to collect interest, and enter 18 as the number of years, when prompted.

Since you have determined to withdraw money at a constant rate, answer no to the question of varying the withdrawal rate. When the program asks you for a constant withdrawal rate, enter 250.

Upon the 1st entry, the program will assemble and then display a summary of your inputs. They should look like this:

Because the most simple option was selected, there is also enough room on the screen for the first 3 years of calculations. Press CONT and ENTER.

Oops! With these assumptions, if the student begins to withdraw money at the start of year 19, he will run out after a little more than 2 years. If you intend the Nest Egg to last for, say, a three year university course, then you will have to change some of your inputs, such as the amount of initial capital or the amount of capital additions.

3) Perpetual Charitable Bequests

Suppose your goal is to make your name live forever by making charitable donations to your favorite cause in perpetuity, based on an initial amount of money that you will bequeath to a trust fund. In order for the donations to be made forever, the annual donations and expenses (or withdrawals) must not exceed the annual income (or interest in this case). Therefore, if you know how much interest will be generated in any year, you automatically know the maximum withdrawal rate for that year.

First, you must estimate the original size of the endowment to the trust fund. Say you intend to bequeath $10,000. When prompted for the initial capital, you would enter 10000. Since you will no longer be around after the initial donation, you answer no to the question of adding more capital in later years. Presumably, you would be doing this estate planning for some future time, several years away. Consequently, you could not possibly make accurate interest rate projections. Therefore, you would probably use a historically average interest of, say, 3%. You would enter no when asked if you wish to vary the interest rate, and 3 when asked for the constant rate.

Since the purpose of the program in this example is to estimate income, you will not be making withdrawal calculations. Therefore, there is no point to using the option of leaving the capital alone to collect interest, without withdrawals. You would enter no when prompted about using this option.

Again, since you will be making no withdrawals, you would enter no, when asked if you would like to vary the withdrawal rate, and enter 0 when asked what constant withdrawal rate you wish to use.

The program will then summarize your inputs, do the calculations, and print the results. But be careful since there are no withdrawals in tha calculations, the calculations would continue until you run out of memory.

From the figures presented, you will be able to determine the income in any year by subtracting the capital left in the year in question, from the capital left in the following year. This income equals the maximum amount that could be given away during that year minus expenses For example the capital left after year 10 is $13,434, and after year 11 is $13,837. Therefore, during year 10, you could begin perpetual withdrawals of $13,837-$13,434, or $403

If you began the withdrawals in year 10, the capital would remain a constant $13,434, because you would be withdrawing the exact amount of interest paid

In this way, "Nest Egg" can be used to determine how long you would have to let the capital collect interest before disbursements of a given size were made from the trust fund

Teaching Value Of The Program

Although "Nest Egg" is flexible and directly applicable to several different situations, its true value is what it can teach about programming the Sinclair to solve everyday problems The main points to be learned are summarized below

Use of GOTO and GOSUB for Calculations

In spite of a potential 250 different combinations of variables in the program, there is only one formula line 1240 This approach is possible because in principle, "Nest Egg" does only one set of tasks it calculates the result of adding income and subtracting withdrawals

The complication is that there are several definitions of both income and withdrawals For example, there is only one interest variable in the formula (Q), but there are potentially 5 different interest rates, or 5 definitions of Q.

In order for the correct definitions of the variable to be selected (e.g., the correct interest rate), it is necessary to initialize 26 variables (lines 2 to 27), use three GOSUBs (lines 1180, 1210, and 1220), use three lines to interchange variables under specified conditions (lines 1190, 1200, and 1230) as well as use a conditional loop (line 1270).

Initializing the variables is required because all variables must be present in the program even if they are not used in the calculations. If a variable has not been initialized by the program or input by the user when running the program, the computer will print an error message and end the running of the program

The three GOSUBs set the conditions for changing the value of the variables in the formula line. For example, Q in line 1240 represents the interest rate GOSUB 3000 changes Q to F, G, H, and I, depending on the year number, and thus introduces varying interest rates into the single line formula. These three GOSUBs are supported by lines 1190, 1200, and 1230 which change the formula variables under conditions not covered by the GOSUBs.

Changing Year Values

The "Nest Egg" faces a problem in telling the computer what year it is from the point of view of each variable. For example, you chose varying interest rates, 3 years of capital accumulation, and varying withdrawal rates Year 6 of the program would be year 6 for interest rates, year 3 for withdrawals and no year at all for interest accumulation without withdrawals.

The problem is to make each year equal to the actual year of the calculations. In this example, the formula must use the first year of withdrawals in year number 4 (i.e., after 3 years of interest accumulation without withdrawals). Thus, the computer must be made to see withdrawal year number 1 as year number 4

This problem was resolved by three methods In the above example, GOSUB 3500 defines the year as the year of the withdrawal *plus* the total years of interest accumulation without withdrawals Thus in the above example, the year number would be 1 (of withdrawals) +3 (of interest accummulation), or year number 4. This approach was used because of the possibility of expressing the relationship between these variables mathematically.

A second method was used for the years of capital additions because no mathematical relationship is possible since any 5 years could be selected with this option. Each of the 5 years was assigned a separate variable value. When the actual year value equals this specific variable value, then the computer seeks the related new capital to add to the existing capital. This process is carried out in GOSUB 2500.

A third method was used for years of interest accumulation without withdrawals, because this option posed yet another problem. The years of this variable are the same as the years of calculations, except that at a predetermined point, the years of interest accumulation cease. Thus, line 1200 removes any reference to this option once the related calculations have been completed. It does so by making the formula withdrawal variable (E) equal to 0, only if it is a year of interest accumulation without withdrawals.

User Friendly

"Nest Egg" is user friendly to the point that separate documentation is not required for its operation. Documentation would be required, however, to fully express the flexibility of the program, and its various applications.

"Nest Egg" takes the user step by step through the selection of options and input of data. A table summarizing the selection of options and inputs is displayed, prior to the actual calculations. The results of the calculations are displayed along with a year number for ease of reference. Finally, the user is given the option of changing the inputs by running the program again. Throughout the program, the only inputs required from the user are numerical inputs and the words "yes" and "no". This user friendliness, however, is memory consuming.

Modifications To Expand The Program

You can modify the program to increase its capacity to meet all possible circumstances posed by the user.

The most obvious change is to increase the number of choices within the

"They never should have come out with this game!"

You can change the progrem by increasing the number of choices in years of capital edditions, in interest rates, and in withdrawal rates.

three variables of 1) new years of capital additions by continuing the WA and XA series of variables beyond the existing 5 to the number desired, 2) varying interest rates, by adding more interest rates, or by changing the number of years during which the rates pertain, and 3) varying withdrawal rates, by adding more rates, or by changing the number of years during which the rates pertain

Such changes would require further changes in the subroutines that govern access to the formula line The latter two changes are not recommended because the degree of accuracy in predicting future interest rates or financial needs, is not sufficient to justify rewriting the program or to ask the user for additional inputs.

The first possible expansion could be handled a better way. The more years that are allowed for new capital additions, the more the option resembles a saving feature So why not add one?

A saving option could be modelled partly on the new capital addition lines and partly on the withdrawal lines Since the user will not have both net withdrawals and net savings in the same years, the only time that net savings can occur is before the withdrawals begin A similar condition has already been created with the option of collecting interest without withdrawals in the first years of the calculations.

The task, then, would be to 1) ask for the year numbers and the monthly savings rates (this input could be userproofed by having the program reject years that exceeded the last year of interest accumulation without withdrawals), 2) assign variable numbers in a similar way to new capital additions inputs, 3) add the savings to the formula calculations in a similar way to the selection of varying withdrawal rates, 4) have a subroutine to change the value of E in the formula to a minus saving rate (E.g., if the savings rate were $200 per month, the subroutine would change E to -200 giving the expression -12*-200, or +2400 In this way, the formula variable intended to represent a withdrawal, in fact becomes a savings.)

This latter approach can be used to modify the original intent of "Nest Egg" without necessarily changing the actual programming. If fewer than 5 years of new capital addition are used, the remaining number of years can be used to add monthly savings. For example, if you have no years of new capital lump sums, but have 5 years of monthly savings, simply multiply the monthly savings by 12 to obtain an annual rate, and enter the data as if it represented lump sum capital additions

Again if you do not use all 5 years for capital additions, you can use the remaining years for lump sum withdrawals. For example, you can introduce the purchase of a new car into the calculations (say in year number 8) by selecting the new capital additions option, and then entering 8 (the year number) and -9000 (the car cost) The minus sign in front of the car price will require the program to treat the entry as a withdrawal, in spite the options original intent of making additions.

Changes made to the actual programming also would require changes in the lines summarizing the user's inputs.

Line Notes:

2-17: Initialize main variables.

40-50: Input initial capital

80-530: Make decision on adding new capital; input new capital additions. Each year has a different variable name (WA to WE) related to the capital amount (XA to XE). After each year's input, the program checks to see if that is the last input to be expected by

```
 BASED ON AN INITIAL CAPITAL OF
$10000

 SOME NEW CAPITAL IN
5 YEARS

 A VARYING INTEREST RATE OF
15 % IN THE FIRST YEAR
12 % IN THE NEXT TWO YEARS
7 % IN THE NEXT THREE YEARS
5 % FOR REMAINING YEARS

 INTEREST GROWTH FOR THE FIRST
5 YEARS

 AND CONSTANT MONTHLY
 WITHDRAWALS OF
$400
```

```
YEAR 1 CAPITAL LEFT $12650
YEAR 2 CAPITAL LEFT $15408
YEAR 3 CAPITAL LEFT $21735
YEAR 4 CAPITAL LEFT $27537
YEAR 5 CAPITAL LEFT $34514
YEAR 6 CAPITAL LEFT $32450
YEAR 7 CAPITAL LEFT $29272
YEAR 8 CAPITAL LEFT $25938
YEAR 9 CAPITAL LEFT $22431
YEAR 10 CAPITAL LEFT $18752
YEAR 11 CAPITAL LEFT $14889
YEAR 12 CAPITAL LEFT $10833
YEAR 13 CAPITAL LEFT $6574
YEAR 14 CAPITAL LEFT $2102
YEAR 15 CAPITAL LEFT $-2593

THAT ENDS THE CALCULATIONS,WOULD
YOU LIKE TO CHANGE ANY OF YOUR
ESTIMATES?
```

```
   1 REM  NEST EGG
   2 LET C=0
   3 LET D=0
   4 LET E=0
   5 LET F=0
   6 LET G=0
   7 LET H=0
   8 LET I=0
   9 LET J=0
  10 LET K=0
  11 LET L=0
  12 LET M=0
  13 LET N=0
  14 LET S=0
  15 LET U=0
  16 LET X=0
  17 LET Y=1
  18 LET WA=0
  19 LET WB=0
  20 LET WC=0
  21 LET WD=0
  22 LET WE=0
  23 LET XA=0
  24 LET XB=0
  25 LET XC=0
  26 LET XD=0
  27 LET XE=0
  30 CLS
  40 PRINT "ENTER INITIAL AMOUNT OF CAPITAL"
  50 INPUT A
  60 CLS
  70 LET B$="YES"
  80 PRINT "WILL THERE BE ANY YEARS IN WHICHYOU INTRODUCE NEW CAPITAL?"
  90 INPUT B$
 100 IF B$<>"YES" THEN GOTO 540
 110 CLS
 120 PRINT "HOW MANY YEARS WILL SEE NEW     ADDITIONS OF CAPITAL? (YOU HAVE A MAXIMUM CHOICE OF 5 YEARS)"
 130 INPUT U
 140 CLS
 150 PRINT "FIRST ENTER THE YEAR NUMBER AT  THE BEGINNING OF WHICH YOU WILL ADD NEW CAPITAL, AND THEN ENTER THE AMOUNT OF NEW CAPITAL."
 160 INPUT WA
 170 INPUT XA
 180 CLS
 190 LET S=S+1
 200 IF S=U THEN GOTO 550
 210 PRINT "YEAR ";WA;" NEW CAPITAL $";XA
 220 PRINT
 230 PRINT
 240 PRINT "ENTER THE NEXT YEAR NUMBER AT  THE BEGINNING OF WHICH YOU WILL ADD NEW CAPITAL, AND THEN ENTER THE AMOUNT OF NEW CAPITAL."
 250 INPUT WB
 260 INPUT XB
 270 CLS
 280 LET S=S+1
 290 IF S=U THEN GOTO 550
 300 PRINT "YEAR ";WB;" NEW CAPITAL $";XB
 310 PRINT
 320 PRINT
 330 PRINT "ENTER THE NEXT YEAR AND THEN     ENTER THE NEW CAPITAL."
 340 INPUT WC
 350 INPUT XC
 360 CLS
 370 LET S=S+1
 380 IF S=U THEN GOTO 550
 390 PRINT "YEAR ";WC;" NEW CAPITAL $";XC
 400 PRINT
 420 PRINT "ENTER THE NEXT YEAR AND THEN     ENTER THE NEW CAPITAL."
 430 INPUT WD
 440 INPUT XD
 450 CLS
 460 LET S=S+1
 470 IF S=U THEN GOTO 550
 480 PRINT "YEAR ";WD;" NEW CAPITAL $";XD
 490 PRINT
 500 PRINT
 510 PRINT "ENTER THE NEXT YEAR AND THEN     ENTER THE NEW CAPITAL."
 520 INPUT WE
 530 INPUT XE
 540 CLS
 550 PRINT "DO YOU WISH TO VARY THE INTERESTRATE OVER THE YEARS?"
 560 INPUT B$
 570 CLS
 580 IF B$<>"YES" THEN GOTO 1500
 590 PRINT "WHAT INTEREST RATE DO YOU WANT  FOR THE FIRST YEAR?"
 600 INPUT F
 610 PRINT
 620 PRINT "WHAT INTEREST RATE FOR THE NEXT 2 YEARS?"
 630 INPUT G
 640 PRINT
 650 PRINT "AND FOR THE SUBSEQUENT 3 YEARS?"
 660 INPUT H
 670 PRINT
 680 PRINT "FINALLY, WHAT SINGLE INTEREST   RATE FOR ALL REMAINING YEARS?"
 690 INPUT I
 700 CLS
 710 PRINT "DO YOU WISH TO ACCUMULATE       INTEREST FOR A NUMBER OF YEARS  BEFORE YOU BEGIN TO WITHDRAW    MONEY?"
 720 INPUT B$
 730 CLS
 740 IF B$<>"YES" THEN GOTO 780
 750 PRINT "HOW MANY YEARS DO YOU WISH TO   ACCUMULATE INTEREST WITHOUT     WITHDRAWLS?"
 760 INPUT J
 770 CLS
 780 PRINT "DO YOU WISH TO VARY THE RATE OF WITHDRAWLS OVER THE YEARS?"
 790 INPUT B$
 800 CLS
 810 IF B$<>"YES" THEN GOTO 2000
 820 CLS
 830 PRINT "HOW MUCH WILL YOU WITHDRAW      MONTHLY IN THE FIRST 2 YEARS?"
 840 PRINT
 850 INPUT K
 860 PRINT "HOW MUCH MONTHLY IN THE NEXT    3 YEARS?"
 870 INPUT L
 880 PRINT
 890 PRINT "AND MONTHLY IN THE NEXT 5 YEARS?"
 900 INPUT M
 910 PRINT
 920 PRINT "AND FINALLY, MONTHLY FOR ALL    REMAINING YEARS?"
 930 INPUT N
 940 CLS
 950 PRINT "  BASED ON AN INITIAL CAPITAL OF$";A
 960 PRINT
 970 IF U>0 THEN PRINT "   SOME NEW CAPITAL IN"
 980 IF U>0 THEN PRINT U;" YEARS"
 990 PRINT
1000 IF C>0 THEN PRINT "  A CONSTANT INTEREST RATE OF     ";C;" %"
1010 IF F>0 THEN PRINT "   A VARYING INTEREST RATE OF    ";F;" % IN THE FIRST YEAR"
1020 IF F>0 THEN PRINT G;" % IN THE NEXT TWO YEARS"
1030 IF F>0 THEN PRINT H;" % IN THE NEXT THREE YEARS"
1040 IF F>0 THEN PRINT I;" % FOR REMAINING YEARS"
1050 PRINT
1060 IF J>0 THEN PRINT "   INTEREST GROWTH FOR THE FIRST   ";J;" YEARS"
1070 PRINT
1080 IF K>0 THEN PRINT "  AND VARYING WITHDRAWLS OF       $";K;" MONTHLY FOR TWO YEARS"
1090 IF K>0 THEN PRINT "$";L;" MONTHLY FOR THREE YEARS"
1100 IF K>0 THEN PRINT "$";M;" MONTHLY FOR FIVE YEARS"
1110 IF K>0 THEN PRINT "$";N;" MONTHLY FOR OTHER YEARS"
1120 IF K=0 THEN PRINT "   AND CONSTANT MONTHLY          WITHDRAWLS OF"
1130 IF K=0 THEN PRINT "$";D
1140 PRINT
1150 PRINT
1160 PRINT
1170 PRINT
1180 IF K>0 THEN GOSUB 3500
1190 IF D>0 THEN LET E=D
1200 IF J>=Y THEN LET E=0
1210 IF U>0 THEN GOSUB 2500
1220 IF F>0 THEN GOSUB 3000
1230 IF C>0 THEN LET B=C
1240 LET A=INT (A*B/100+A-12*E)
1250 PRINT "YEAR ";Y;" CAPITAL LEFT $";A
1260 LET Y=Y+1
1270 IF A>0 THEN GOTO 1180
1280 PRINT
1290 PRINT
1300 PRINT
1310 PRINT
1320 PRINT "THAT ENDS THE CALCULATIONS,WOULDYOU LIKE TO CHANGE ANY OF YOUR  ESTIMATES?"
1330 INPUT B$
1340 IF B$="YES" THEN GOTO 2
1350 PRINT
1360 PRINT
1370 PRINT "OK, THAT ENDS THE RUNNING OF THEPROGRAM. TRY IT AGAIN IF YOU    HAVE MORE QUESTIONS."
1380 STOP
1500 PRINT "WHAT CONSTANT INTEREST RATE     WOULD YOU LIKE TO USE?"
1510 INPUT C
1520 GOTO 700
2000 PRINT "WHAT CONSTANT AMOUNT DO YOU WISHTO WITHDRAW EVERY MONTH?"
2010 INPUT D
2020 GOTO 940
2500 IF Y=WA THEN LET X=XA
2510 IF Y=WB THEN LET X=XB
2520 IF Y=WC THEN LET X=XC
2530 IF Y=WD THEN LET X=XD
2540 IF Y=WE THEN LET X=XE
2550 IF Y=WA OR Y=WB OR Y=WC OR Y=WD OR Y=WE THEN LET A=A+X
2560 RETURN
3000 IF Y=1 THEN LET B=F
3010 IF Y=2 OR Y=3 THEN LET B=G
3020 IF Y=4 OR Y=5 OR Y=6 THEN LET B=H
3030 IF Y>6 THEN LET B=I
3040 RETURN
3500 IF Y=1+J OR Y=2+J THEN LET E=K
3510 IF Y=3+J OR Y=4+J OR Y=5+J THEN LET E=L
3520 IF Y=6+J OR Y=7+J OR Y=8+J OR Y=9+J OR Y=10+J THEN LET E=M
3530 IF Y>10+J THEN LET E=N
3540 RETURN
```

Now...Reston has a complete line of software and books for the Timex Sinclair & ZX-81 computers.

RESTON COMPUTER GROUP
Reston Publishing Company, Inc.

SoftWare

- ☐ R1261-6 **Demolition/Ten Pin** — Two action packed games in one package! $9.95
- ☐ R4608-7 **Monarch** — Control land! Rule People! See how long you can keep the throne! $12.95
- ☐ R1751-7 **Escape from Shazzar!** — Find the temple and escape the evil that lurks around every corner! $12.95
- ☐ R0705-5 **Casino Pak I: One-Arm Bandit and Blackjack** — Take a trip to Las Vegas! Play against your computer and see who comes out the big winner! $14.95
- ☐ R4953-7 **The Nowotnik Puzzle** — Can you unscramble the puzzle? It's harder than it looks! $14.95
- ☐ R0738-4 **Championship Chess** — Good news for chess players! Your computer challenges you to this award winning game! $14.95
- ☐ R6968-7 **Self-Teaching Software for the Timex Sinclair 1000— Level 1** $29.95
- ☐ R6967-5 **Self-Teaching Software for the Timex Sinclair 1000— Level 2** $29.95
- ☐ R6968-3 **Self-Teaching Software for the Timex Sinclair 1000— Level 3** $29.95
- ☐ R6970-9 **Self-Teaching Software for the Timex Sinclair 1000— Level 4** $29.95
- ☐ R6971-7 **Self-Teaching Software for the Timex Sinclair 1000— Level 5** $29.95
- ☐ R6972-5 **Self-Teaching Software for the Timex Sinclair 1000— Level 6** $29.95

BookWare

- ☐ R7697-7 **The Timex Sinclair 1500/1000 Pocket Book** — A complete handbook for the most popular "little" computer around! $9.95
- ☐ R0348-4 **BASIC Basics for the Timex Sinclair 1500/1000** — Learn BASIC and discover the power of the computer. $9.95
- ☐ R1890-3 **50 1K/2K Games for the Timex Sinclair 1500/1000** — Dogfight! Ski Run! UFO Attack! Programming instructions turn your computer into a game arcade! $9.95
- ☐ R0347-6 **Basics of Timex Sinclair 1500/1000 BASIC** — A wonderful introduction to the world of BASIC! $9.95
- ☐ R2088-7 **49 Explosive Games for the Timex Sinclair 1500/1000** — Galactic Intruders, Checkers, Smuggler's Gold — Here are programming instructions for your favorite games! $9.95
- ☐ R4255-7 **Mastering Machine Code on Your Timex Sinclair 1500/1000** — Finally a "simple English" guide to machine code! $9.95
- ☐ R4190-6 **Making the Most of Your Timex Sinclair 1500/1000** — A great way to learn about each feature and function of your computer! $9.95
- ☐ R1816-9 **Explorer's Guide to the Timex Sinclair 1500/1000** — Discover the limitless capabilities of your computer. $9.95

☐ Check enclosed ☐ VISA ☐ MasterCard

Name ______________________
Address ______________________
City ____________ State ______ Zip ________
Card # ______________________
Interbank # ____________ Expires ________
Signature ______________________

For the Timex Sinclair 1000 Series and ZX-81 Computers / Available from your local bookstore OR Timex Sinclair Computer Outlet

Reston Publishing Company, Inc., Attn: Marketing, 11480 Sunset Hills Road, Reston, VA 22090 **Or call toll-free** 800-336-0338

TIMEX SINCLAIR is a registered trademark of TIMEX Corp.

CIRCLE 42 ON READER SERVICE CARD

ZX-81 is a registered trademark of Sinclair Research LTD comparing variable W (number of years of new capital additions) to the variable S (the number of years input so far)

550-690 Make decision on varying interest rates, input rates

710-760 Make decision on accumulating interest without withdrawals, input number of such years.

780-930 Make decision on varying the withdrawal rates, input the rates.

950-1130 Display a summary of the options and inputs. Note that the printing of any line is governed by a conditional statement which defines whether user has selected the related option. E.g., lines 1010-1040 are printed only if the variable F is greater than O. Since F is the first of the four varying interest rates, it must be greater than O only if the user selected the option of varying the rates. If the user selected the constant rate, these lines would not be printed

1180-1230 Introduce the subroutines and conditional statements for selecting the correct variables for inclusion in the calculation formula.

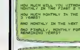
1240 The formula line: A = capital, Q/100 = interest rate, and 12*E = 12 times the monthly withdrawal rate. The subroutines and conditional statements substitute variables within the formula, but do not change the formula itself. Variable A is changed by subroutine 2500, Q by 3000 and the conditional statement in 1230; E by 3500 and the conditional statements in 1190 and 1200.

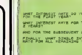
1250-1270 Print the results of the calculations, return the program to 1180 to calculate the next year; the process continues until the capital is exhausted, i.e., until A is no longer greater than O.

1500-1520 Input the constant interest rate; program goes to the line following the last varying interest rate line.

2000-2020 Input constant withdrawal rate, program goes to line following the last varying withdrawal rate line.

2500-2560 Select amounts of new capital to be added to the initial capital depending on the year number; change the value of the related variable A in formula line 1240





3000-3540 Select any of four withdrawal rates depending on the year number and the number of years (J) allowed for interest to accumulate without withdrawals. Thus LET Y = J + 1 means, in effect, that, if the year of the calculation equals the number of years without withdrawals plus 1, you should use the withdrawal rate applicable in year 1. This process changes the value of the related variable E in line 1240.

A copy of "Nest Egg" is available on cassette for $9.95 from the author

Split and Save *Harold Miller*

Would you like to freeze the upper part of your display screen? With "Split 'n Save" you can! While the information in the lower portion is appearing and disappearing, any text or graphics in the upper portion remain on view.

For a demonstration of "Split 'n Save," enter and RUN the program in Listing 1.

The heart of this program is the subroutine beginning at line 900, which adds as many enter codes to the end of the last frozen display line as there are lines being frozen. This tricks the computer into thinking that display line 0 is loaded immediately below the frozen zone, rather than at the top of the screen. Now CLS, PRINT AT, PLOT and UNPLOT will act accordingly! For example, make this alteration and see what happens.

110 PRINT "THIS"

To add "Split 'n Save" to your program, insert the following lines.

DIM A$(N)

N is the number of frozen display lines

PRINT AT N,O, A$

This reserves N spaces in the last frozen display line. If your program is PRINTing on this line, you can add A$ to it. (See line 90 in Listing 1.) In any case, the total number of display characters must not exceed 32 (including the N spaces).

GOUSUB 900

This turns on the "Split 'n Save" feature. Place it immediately after the program line in step (2).

GOSUB 1000

This turns off the "Split 'n Save" feature.

Also, place both subroutines at the end of your program.

"Split 'n Save" has one peculiarity: display line 0 is not at the top of the screen, so lines 22 and 23, which contain any report, INPUT symbol or INPUT Data, are not visible.

Line Notes

40 Clears screen before freezing.
50 Creates N spaces.
60 Loop prints in frozen zone only.
90 Adds N spaces to end of last frozen line.
100 "Split 'n Save" on.
110-210 Display function will not affect frozen zone.
230. "Split 'n Save" off.
240 Clears entire screen.
900 Location in DFILE of the first of the N spaces.
910 Loop replaces each space with ENTER.
1000 Must clear lower part of screen before turning off "Split 'n Save."
1010. Loop to reset each of the N extra ENTERs to zero.

Listing 1. Split 'n Save Demonstration.

```
  10 REM SPLIT N SAVE DEMO
  20 PRINT AT 20,0, NO. OF FROZE
N DISPLAY LINES?     (MAX=10)
  30 INPUT N
  40 CLS
  50 DIM A$(N)
  60 FOR L=0 TO N-2
  70 PRINT  FROZEN LINE  ,L
  80 NEXT L
  90 PRINT  FROZEN LINE  ,N-1,A$
 100 GOSUB 900
 110 PRINT AT 0,0, THIS
 120 PRINT  SMALL
 130 PRINT  PASS
 140 PRINT
 150 PRINT  (HIT ENTER)
 160 PAUSE 1000
 170 CLS
 180 PRINT AT 3,0, THIS TOO SMAL
L PASS (HIT ENTER)
 190 PAUSE 1000
 200 CLS
 210 PRINT  HIT ENTER TO CLEAR E
NTIRE SCREEN
 220 PAUSE 1000
 230 GOSUB 1000
 240 CLS
 250 STOP
 900 LET S=PEEK 16400+256*PEEK 1
6401-65
 910 FOR I=0 TO N-1
 920 POKE S+I 118
 930 NEXT I
 940 RETURN
1000 CLS
1010 FOR I=0 TO N-1
1020 POKE S+I,0
1030 NEXT I
1040 RETURN
```

Harold Miller, Rt. 2, Box 2330, Clayton, GA 30525

Chaining Programs

Harold Miller

When one program is on the computer and one is on tape, we need a MERGE command, but the 8K ROM does not have it.

The Problem

Prom time to time we have two programs or more that we want to combine—perhaps a main program and a subroutine. Obviously, this can be done if they are both typed in from scratch.

However, what do we do when one program is on the computer and one is on tape or when both are on tape? If we attempt to LOAD a program from tape into the computer when it already has a program resident, we find that it will erase the first program. We could type in the second program, but we certainly would like to avoid that much work. Some computers have a MERGE command to accomplish this, but the 8K ROM does not include a MERGE command.

The ZX/TS Solution

On the ZX/TS computers, however, we do have an option: we can "chain" the programs by the method developed in this article so that a program on tape can be LOADed in the computer when a program is already in RAM.

The method in outline is: 1) Protect the resident program from erasure by POKEing a copy of it above RAMTOP. This area of memory, once created, is unaffected by LOAD, SAVE, and NEW. 2) Now LOAD the other program from the tape in the normal manner. 3) POKE the original program into the program area immediately above the LOADed program. Now you have access to both programs which can be RUN, SAVEd and EDITed.

Now let's apply the method. Suppose we have two programs, A and B, which we want to chain together with B following A. We will illustrate the process for the TS1000 with the Sample Programs A and B below.

Harold Miller, Rt. 2, Box 2330, Clayton, GA 30525.

Steps in Chaining

The following steps will enable you to chain A and B.

Step 1. Check the line numbers

Before chaining, you must be sure that none of the line numbers in Program B are used in Program A because a line already on the computer with a given line number is replaced by a new line entered with the same number. If you plan ahead and assign relatively high numbers in Program B, you can avoid having to renumber the lines in Program A. You may want to use a line renumber utility to save editing the line numbers.

Step 2. LOAD (or ENTER) Program B

We must modify Program B so that whenever it is LOADed, a copy is automatically placed (POKEd) into a protected region of memory, i.e., above RAMTOP, so that it will not be cleared when Program A is entered or LOADed. This is done through steps 3-6 below, but when it has been done once, it never has to be repeated.

Sample Program B

```
1000 PRINT 1
1010 PRINT 2
1020 PRINT 3
1030 LET A=4
```

Step 3. Determine the program size

Since the program area begins at 16509 and ends at D-FILE-1 (see the chapter on the organization of memory in your manual), you can determine size by typing in the immediate mode (without a line number):

```
PRINT PEEK 16396+256*PEEK 16397-16509
```

This yields the number of bytes in the program. Let us call this number *n*. *Note:* All italicized letters in lines and listings below must be replaced by their appropriate values prior to execution. For Sample Program B, n = 54.

Step 4. Calculate the new RAMTOP

Begin with the RAMTOP value for your system: 1K = 17408; 2K = 18432; 16K = 32768. Subtract *n*. Let's call the answer *r*. If you have 64K, omit this calculation and use *r* = 8192. For Sample Program B, *r* = 18378.

Step 5. Modify Program B

To Program B add, with the appropriate line numbers, the following:

```
STOP
FOR I=0 TO n-1 (line y)
POKE (r+I),PEEK (16509+I)
NEXT I
STOP
```

Replace n-*1* in line *y* and *r* with the appropriate values. For Sample Program B, this would give:

```
2000 STOP
2010 FOR I=0 TO 53
2020 POKE (18378+I),PEEK (16509+I)
2030 NEXT I
2040 STOP
```

Step 6. SAVE Program B.

Save this program, e.g., SAVE "BB".

Step 7. Adjust RAMTOP

Move RAMTOP down *n* bytes by the following steps. (If you have 64K, omit this step.)

a) After replacing *n* with the appropriate value and carrying out the calculations, enter the following line in the immediate mode:

```
POKE 16388,256*(1+INT (n/256))-n
```

For Sample Program B, this gives

POKE 16388,202

b) After replacing n and z with the appropriate values (z = 67 for 1K, 71 for 2K, 127 for 16K) and carrying out the calculations, enter this line in the immediate mode:

POKE 16389,z-INT (n/256)

For Sample Program B, this gives

POKE 16389,71

c) Press NEW and ENTER. While this erases whatever was in memory, it also moves RAMTOP down n bytes.

Step 8 Put Program B above RAMTOP

LOAD and RUN the routine at line y in the modified version of Program B, e.g., LOAD "BB" and RUN 2010. A copy of Program B is now above RAMTOP and cannot be erased by subsequent LOADs, SAVEs, or NEWs.

Step 9 LOAD Program A

LOAD Program A from tape (if you are entering Program A, hit NEW and ENTER to get rid of Program B).

Sample Program A

```
100 PRINT "A"
200 PRINT "B"
300 PRINT "C"
400 LET B=4
```

Step 10 Determine the size of Program A

Determine the number of bytes in Program A just as you did with Program B in Step 3 above. Let's call this number m. For Sample Program A, m = 42.

Step 11 Adjust Program A

Add the following short routine to Program A, replacing n-1, m, and r by their appropriate values and adding the appropriate line numbers:

```
STOP
FOR I=0 TO n-1 (line z)
POKE (16590+m+I),PEEK (r+I)
NEXT I
STOP
REM (type in n-6 A's)
```

This REM line sets aside a total of n bytes for Program B to fit into. The value 16590 assumes that n is two digits, i.e., n is between 10 and 99. For each additional digit in n add 1 to 16590. If you expect to chain these two programs again, SAVE this modified version of Program A, e.g., SAVE "AA". For Sample Program A the lines would be

```
500 STOP
550 FOR I=0 TO 53
600 POKE (16632+I),PEEK (18738+I)
650 NEXT I
700 STOP
800 REM (type in 48 A's)
```

If your Program B is very large, you might want to use the technique discussed by Jasper Kump (*SYNC* 3:2, p. 60) to reduce the drudgery.

Step 12 Execute the chaining routine

Type in

GOTO line z

This replaces the REM line in Program A with Program B. In our example this would be

GOTO 550

Step 13 Delete the separating lines

Delete the remaining lines that separate Programs A and B. In our example, delete lines 500, 550, 600, 650, 700.

Step 14 Restoring RAMTOP (Optional).

If you are short on memory and no longer need the copy of Program B above RAMTOP, SAVE your chained program, and return RAMTOP to its original location by unplugging the computer. Then turn on the computer and LOAD the chained program again.

Conclusion

This process can be extended to chain as many programs as RAM space permits. Also, since the modified versions of the programs have been SAVEd, chaining A and B on subsequent occasions is greatly simplified:

1) Reserve n bytes above RAMTOP.
2) LOAD "BB"
3) LOAD "AA"
4) GOTO z
5) Delete the 5 unwanted lines

Bit by Bit *Harry Doakes*

10101101
↓
10100101

The computer sees every number as a string of just 8 bits; it can look at and change just one bit by TEST, SET, and RESET.

When we use numbers in a computer, we usually think about them as whole numbers Generally we do not think of them as bytes, and we certainly do not think about the fact that everything in the computer is ultimately a bunch of bits

This time we will take a look at bits—the smallest pieces of information a computer can handle—and how the Z80 processor in your ZX/TS computer can manipulate them. You may never need most of these instructions, but it is a good idea to know something about them anyway. Then we will do something very different—we will look at how to use the Zilog *Z80 CPU Programmer's Reference Guide.*

Bit Wise

What is a bit?

Bit is one of the oldest pieces of computer jargon It is short for "binary digit" —just the first and last letters, pushed together A *digit* is one of the numerals that make up a number The value of a number depends on what the digits are, and what position each digit is in. *Binary* means the number is in base 2. In base 10—the decimal counting system we normally use—we have 10 different digits. 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 In base 2, we have only 2 different digits 0 and 1 Thus, the first few numbers in binary look like this 1, 10, 11, 100, 101, 110, 111, and 1000.

We already know that a regular Z80 register or a single byte of memory can hold a number from 0 to 255 decimal. In base 2, that is between 0 and 11111111. Notice that the number 11111111 is 8 binary digits long The Z80 is called an *8-bit processor* because the largest value its regular registers can hold is 8 bits long

Your manual has a chapter on counting in both binary and hexadecimal, and why these number systems are important to understand when you are programming a computer

One Bit at a Time

Up until now we have mainly dealt with numbers in the computer the way we see them, i e , as whole numbers

But the computer sees numbers differently. Every number stored in the computer—in ROM, RAM or a register—is just 8 bits to the computer. Think about that a moment It is important. Since the computer sees each number just as a string of bits, there is no reason it cannot look at, or even change, just one bit of a number, leaving the other 7 bits alone

In fact, the Z80 processor does so by using the instructions *test, set,* and *reset*

TEST

The *test* instruction, BIT, tests one bit For example, suppose that register A contains the number 45d It would look like this

binary 00101101
bit number 76543210

The 8 bits are numbered 0 to 7, bit 7 is on the far left with bit 0 on the far right

Now suppose your machine code program uses the instruction

BIT 4,A

This means "test bit 4 of register A." Bit 4 is the *fifth* bit from the right, remember, the first bit is called "bit 0" As you can see, there is a zero in bit 4. Consequently, the zero flag of the Z80 will go up

On the other hand, suppose you use the instruction

BIT 3,A

Bit 3 is not 0, but 1. So the zero flag would come down.

You can use the BIT instruction with jump instructions such as JP Z or JR Z to make your program do different things depending on whether a bit is a 0 or a 1 For example,

BIT 4,A
JR Z,branch

would be something like this in Basic

IF (BIT 4 OF A)=0 THEN GOTO 240

Of course, you cannot say that in Basic since there is no "BIT" function But you *can* do it in machine code.

SET

You can also force a particular bit of a particular register to become either 1 or 0 The *set* instruction, SET, will set the bit to 1, the *reset* instruction, RES, resets it to 0 For example, suppose register A is 45d In binary it looks like this

00101101

Now, if you used the instruction

SET 7,A

register A would have a value of

10101101

or 173d The difference is that bit 7, on the far left side, has been set to 1.

Then if you used the instruction

RES 3,A

register A would look like this

10100101

Bit 3 has now become a zero and register A has a value of 165d

Get the idea? SET makes sure that the bit is 1. RES makes sure that the bit is 0 That *will not always* change the value of

Harry Doakes PO Box 10860 Chicago IL 60610

the number, if you try to set a bit that is already 1, or reset a bit that is already 0, nothing will happen. However if you do a SET or RES, you will know for sure what the bit will be

What good is knowing that? Well, sometimes it is critical that a certain bit be 0 or 1 For example, when you use the ROM subroutine CALL 16, the character whose value is in register A will be printed on the screen However, that must be a number between 0 and 63 or 128 and 191 (the only exception is the ENTER code, 118). These numbers all have one thing in common bit 6 is always zero. If bit 6 is 1, the system will crash To keep that from happening, you could use the instruction

RES 6,A

before you

CALL 16

Then you would know for sure that bit 6 is zero and that the CALL will not crash the system.

BIT, SET, and RES can be used with any bit of any of the regular-size registers A, B, C, D, E, H, or L. As usual, you can also use register pair HL as a pointer to a byte in memory with these instructions

Shifting Gears

You can also *move* bits in a register. That should not be a surprise After all, to the Z80, a number is just a line of bits. You can shift them down through a register in one direction or the other with a *shift* instruction Or, if you like, you can run the bits around in a circle with a *rotate* instruction

We have already encountered the shift instruction in dividing and multiplying by 2. A *shift left* instruction (SLA) moves all the bits to the left one position For example, suppose that register C holds 203 In binary it looks like this.

11001011

After a "shift register C left" instruction (SLA C) it would look like this

10010110

Notice that all the bits moved one position to the left, and a zero was added on the far right side at bit 0 The original bit 7, on the far left side, seems to have disappeared But it is not really gone In case you need that bit, the carry flag keeps track of it If the bit was 1, the carry flag goes up, if the flag was 0, the carry flag goes down The process looks something like this

carry flag ←— register ←— 0

In short, a zero is pushed in on the right side, and the left bit drops into the carry flag If the number is less than 128, this instruction has the effect of multiplying it by 2

There are two "shift right" instructions that can be used to divide a number by 2. The first is the "shift right logical" (SRL) It pushes a zero in on the far left at bit 7, bit 0, on the right, drops into the carry flag, like this

0 —→ register —→ carry flag

There is also a "shift right arithmetic" instruction (SRA) This one works a bit differently instead of pushing a zero into bit 7, it just leaves it alone However, it also shifts everything down, and bit 0 once again drops into the carry flag This instruction is used because sometimes bit 7 is used as a "sign" bit When that is the case, it is important that bit 7 stay the same when the number is shifted

↳register —→ carry flag

The Circle Game

Now comes the interesting part getting the registers of the Z80 to go around in circles You can do this with one of the four "rotate" instructions.

"Rotate left" (RL) works like this

1) The carry flag is pushed into the right side of the register at bit 0

2) Everything moves over one bit

3) Bit 7 drops into the carry flag

It looks something like this

carry flag ←— register ←— carry flag

"Rotate right" (RR) is just the opposite:

carry flag —→ register —→ carry flag

As you can see, the bits actually do go in a circle If, e g., you did nine RR H instructions in a row, register H would end up exactly as it had started But it does take 9 not 8, rotates, since the circle includes the carry flag

You can tighten up the circle by using the "rotate left circular" (RLC) and "rotate right circular" (RRC) instructions These instructions actually rotate just the eight bits of the register in a circle RLC moves everything to the left, and bit 7 becomes the new bit 0 RRC moves everything to the right, bit 0 becomes the new bit 7 In each case, the bit that gets bumped to the opposite end is also copied into the carry flag If the bit is 1, the carry flag goes up, if the bit is 0, the flag comes down

Perhaps this all sounds interesting, but pretty useless Why would anyone want to run the bits of a register in a circle?

Actually, these instructions are usually used with register pairs to multiply or divide by 2 For example, you can divide a number in register pair BC by 2 with these instructions

SRL B
RR C

Here is what happens the "shift right logical" instruction shifts all the bits in register B to the right, dividing it by 2. Bit 0, of course, gets pushed into the carry flag But we want that bit to become the new bit 7 of register C So the "rotate right" instruction gets that bit from the carry flag and pushes it into the left side

OR A will not change the value in register A, but it guerentees that the carry fleg is down.

of register C, moving everything else to the right and dividing *it* by 2 It looks something like this

```
0 → register B → carry flag
carry flag → register C → carry flag
```

You can multiply register pair BC by 2 with this set of instructions

```
SLA C
RL B
```

Once again, you can use these instructions with any of the regular 8-bit registers, or by using register pair HL as a pointer to a byte in memory.

Let's Get Logical

We already know about the AOO and SUB instructions, which let you add or subtract a number with what is in register A Along with these "arithmetic" instructions, there are also three "logical" instructions: AND, OR, and XOR

Each of these instructions compares a number with register A, one bit at a time The two values in bit 0 are compared, and the result goes in bit 0 of register A Then the bit 1s are compared, the result going into bit 1, then the bit 2s are compared, and so on through bit 7 Exactly what the result of each comparison is depends on which logical instruction is being used

For example when AND compares two bits the result is 1 only if both bits are 1. If either bit is 0, the result is 0

When OR compares two bits, the result is 0 only if both bits are 0 If either is 1, the result is 1

XOR, sometimes called the "exclusive-OR," is a little different When XOR compares two bits, if the bits are the *same*, the result is 0; if the bits are *different* the result is 1

A few examples will show how these operations work.

AND is often used to find out whether a particular bit is 1 or 0 For example,

```
register A 10010100
ANO 00000100
result 00000100
```

You can use the AND instruction, then a jump depending on whether bit 2 is one

```
AND 4
JR NZ, branch
```

However, you can do the same thing with a BIT instruction

```
BIT 2,A
JR NZ, brauch
```

ANO can also be used to make sure a particular bit is zero For example,

```
register A 10010010
AND 11111101
result 10010000
```

You can usually do the same thing with an RES instruction

OR can be used to make sure a particular bit is 1 For example

```
register A 10010010
OR 00000100
result 10010110
```

Of course, you can get the same result with a SET instruction.

XOR is really the only one of these three instructions that does something unique it can "flip" a bit For example,

```
register A 10010010
XOR 10000000
result 1 00010010
XOR 10000000
result 2 10010010
```

Notice how bit 7 has been reset in result 1, then set again in result 2 To do this using SET and RES instructions you would have to do something like Figure 1

There is one other thing you should know about the logical instructions no matter what the result is, they always bring down the carry flag Whenever you

need to make sure the carry flag is down, you can use an instruction such as

OR A

It will not change the value in register A (can you figure out why?); it just guarantees that the carry flag is down.

Flashdance

A few months back, we put together a program that reversed everything on the screen. To demonstrate how the XOR instruction can be used, this time we will make the screen "flash" by switching what is on the screen from regular characters to reversed characters and then back again. The routine also demonstrates how you can put a "pause" in your machine code programs—just as you can in Basic.

The program is in Figure 2. Since it is such a short, simple routine, there are no machine code variables. Register C keeps track of how many times the screen has been reversed. It starts at 2 and then drops by 1 each time through. When it reaches zero, the routine returns to Basic.

The section from the line labeled "start" down to "delay" is the routine that reverses the screen. It checks each byte, beginning at the start of the display file. If the byte is a regular character, it is reversed using the instruction

XOR 128

This "flips" bit 7 of the character. (If bit 7 is a zero, it is a regular character; when bit 7 is 1, the character is reversed.)

If the character is an ENTER (character code 118) it is not XORed; instead, register B is decremented. Register B starts out at 22; after all 22 lines on the screen have been reversed, register B is 0, and the reversing routine is finished.

Take a careful look at the section starting at the line labeled "delay." This is the "pause" routine. Here is how it works: 60 times each second the computer sends a picture to your TV screen, andf, every time it does, it reduces by one the system variable called FRAMES. However, the top bit of this double-sized variable is always kept as 1, so that FRAMES is always a number between 32768 and 65535.

At "delay" we put FRAMES in register pair HL, and put 15 in register pair DE. Then we subtract DE from HL with the instructions

```
OR A
SBC HL,DE
```

The SBC instruction (SuBtract with Carry) will subtract DE from HL—but it will also subtract *one more* from HL if the carry flag is up. That can be useful sometimes, but not here. We use OR A to make sure the carry flag is down, so we know we will get the correct answer to HL minus DE.

Next we SET bit 7 of register H. Now HL holds the same number that FRAMES will hold once it has sent 15 more pictures to the TV screen, i.e., in about 1/4 second. We move that number to register pair DE.

Finally, in the section labeled "loop" the program compares the number in DE to FRAMES again and again. It keeps jumping back to "loop" until at last they are the same; then it goes on to decrement register C and either reverse the screen again or return to Basic. You can use the

Figure 1

```
        BIT 7,A       ;test the bit
        JR Z,skip1    ;if it is 0, jump to "skip1"
        RES 7,A       ;if it is 1, make it 0
        JR skip2      ;jump to "skip2"
skip1:  SET 7,A       ;if it is 0, make it 1
skip2:  ...           ;(next instruction)
```

Figure 2.

```
0E 02          LD C,2         ;reverse screen twice
06 16   START: LD B,22        ;22 lines
2A 0C 40       LD HL,(D-FILE) ;start of display file
23      TOP:   INC HL         ;next character
7E             LD A,(HL)      ;get it
FE 76          CP 118         ;is it an ENTER?
28 05          JR Z,line      ;if so, skip ahead
EE 80          XOR 128        ;reverse character
77             LD (HL),A      ;replace the character
18 F5          JR top         ;do it again
05      LINE:  DEC B          ;next line
20 F2          JR NZ,top      ;if less then 22 lines
2A 34 40 DELAY: LD HL,(FRAMES) ;current FRAMES
11 0F 00       LD DE,15       ;pause time
B7             XOR A          ;reset carry flag
ED 52          SBC HL,DE      ;HL=HL-DE
CB FC          SET 7,H        ;to match FRAMES
54             LD D,H         ;DE=HL
5D             LD E,L
2A 34 40 LOOP: LD HL,(FRAMES) ;check till they match
AF             XOR A
ED 52          SBC HL,DE
20 FB          JR NZ,loop
0D             DEC C          ;finished reversing?
20 D5          JR NZ,start    ;if not, jump
C9             RET            ;otherwise, return
```

Figure 3

First RUN this program to reserve high memory. The "flasher" program takes 46 bytes

```
 10 PRINT "HOW MANY BYTES?"
 20 INPUT A
 30 LET RT=PEEK 16388+256*PEEK
16389
 40 LET RT=RT-A
 50 LET H=INT (RT/256)
 60 LET L=RT-256*H
 70 POKE 16388,L
 80 POKE 16389,H
 90 NEW
```

Then enter and RUN this program to load the flasher" routine into high RAM

```
  1 REM 0E0206162A0C40237EFE762
805EE807718F50520F22A3440110F00B
7ED52CBFC545D2A344087ED5220FB0D2
0D5C9
 10 LET RSTART=16514
 20 LET START=PEEK 16388+256*PE
EK 16389
 30 LET A=0
 40 LET H=PEEK (RSTART+2*A)-28
 50 IF H<0 OR H>15 THEN STOP
 60 LET L=PEEK (RSTART+2*A+1)-2
8
 70 IF L<0 OR L>15 THEN STOP
 80 LET N=16*H+L
 90 POKE START+A,N
100 LET A=A+1
110 GOTO 40
```

"pause" routine with any machine code program Depending on what value you load into DE, your program can pause as little as 1/60 of a second or as long as 9 minutes Since machine code works so fast, sometimes that is e very good thing to have

Use the Basic programs in Figure 3 to load the "flasher" routine into high memory, then try it out with a short program such as in Figure 4 It will work in as little as 1K RAM, but it does need the 8K ROM

Figure 4.

```
 10 LIST
 20 LET START=PEEK 16388+256*PE
EK 16389
 30 LET A=USR START
```

Otherwise, you will not be able to see the screen flash.

The Guided Tour

By now, if you have been following this series of articles, you know something about most of the instructions of the Z80 microprocessor We have covered such things as registers and flags, loading and jumping, calls and compares. And until now, I have explained what each machine code instruction does and how it works as we encountered it.

Now that you have a good grasp of what machine code is, it is time to take the next step to explore the instructions of the Z80 in detail We will do that with a little book that is packed with all sorts of technical information *The Z80 CPU Programmer's Reference Guide.* You can get a free copy of the book from Zilog, the company that designed the Z80, by writing to Zilog, 1315 Dell Avenue, Campbell, CA 95008 Be sure to mark your letter "Attn Tech Publications" and ask for the guide by its full name

The guide is filled with charts, diagrams, and explanations, and—as we will see—a few surprises as well.

More Registers?

The surprises start on page 2 of the guide, with a diagram of the registers of the Z80 Some of the registers look familiar In the "main register set" we can spot A, B, C, D, E, H, and L, along with the flags register, register F. But you will notice that, right next to that diagram, there is another set of registers that looks just like the first one—the "alternate register set."

These registers are just exactly like the first set. This is a "spare" set. Like a spare tire, it can come in handy for a programmer, though usually you can get by without it. When your program uses the "exchange registers" instruction,

EXX

it is as if you have jacked up your Z80 processor, taken off registers B, C, D, E, H, and L, and put on the spare registers. To get the original registers back, you just use another EXX instruction.

But it is not a good idea to use these spare registers in your machine code programs. The Basic interpreter program in your computer uses some of these registers to keep track of what is going on in Basic. Using the spare registers may crash the computer or make your Basic programs act very strangely.

You can also exchange register A and the flags register with their spares. That happens if you use the "exchange registers A and F'" instruction,

EX AF,AF'

—but *do not do it!*

Why not? Because the Timex and Sinclair computers use the alternate register A for putting things on your TV screen. Sixty times every second when you are in "slow" mode, your ZX81 or TS1000 stops what it is doing and goes to work sending information to the TV. The TV will not wait, so the computer does not have time to check on register A, or do any other kinds of "housekeeping."

Do not use the alternate registers, or the special-purpose registers I, R, IX, or IY, for any machine code routines on the Sinclair and Timex computers. If you do, you will probably crash the system. That is right—you should not use any of the new registers on page 2. That is why we have not looked at them before. For programming on the ZX80, ZX81, and TS1000, they might as well not exist.

Hot Off the Grid

On pages 4 and 5 you will see some familiar information in a new form. Page 4 contains a grid for instructions in the "8-bit Load Group." That means they are load instructions that move numbers between regular-sized registers or single bytes of memory. (Remember, the Z80 is called an 8-bit processor because its regular registers handle 8 bits at a time.)

You can use the grid to translate an instruction into the numerical code that the Z80 processor understands. Along the top side of the "8-Bit Load Group" grid are the names of "sources"—the registers and places in memory that a number can come from. Down the left side are the possible "destinations"—places a number can go. To use the grid, you simply choose a load instruction—a source and a destination. Then find the place where the column of the source and the row of the destination meet. That is where you will find the hexadecimal number that matches your instruction.

For example, let's try the instruction

LD A,B

This loads the value in register B into register A. Register B is the source, register A is the destination. Follow the "B" column down and the "A" column across, and you will see that they meet at the number 78. (78 is in hex, or base 16.)

Now turn to the "Character Set" appendix in the back of your ZX81 or TS1000 manual. Look down the column headed "Hex" until you find the number 78. You will find that 78h corresponds to 120d—and, just as expected, it also corresponds to the machine code instruction LD A,B.

The appendix and the grid both contain the same information. The appendix is organized with the instructions in numerical order, the guide, according to what each instruction does. Each is handy to have, though most people find it easier to use the grid for translating instructions into the numerical codes.

Partial Pascal

Bring Tape Under Control

Partial Pascal makes tape easier and more convenient to use The *write* statement in your Partial Pascal program stores output to tape in buffers in RAM When a buffer is full (or when your program indicates it has no more data to write), Partial Pascal asks what name you want to give it and records a header, the name you specify and the data from the buffer, releasing it to be refilled with more data from your program

Any Partial Pascal program can read data written to tape by the editor or any other Partial Pascal program When your program has to read data from a tape file, Partial Pascal asks you for the name you gave the data when it was recorded Partial Pascal finds the data with the name you've asked for, reads it into a buffer and provides it to your program from the buffer

Partial Pascal is not fussy at all about positioning the tape for input Anywhere before the header of the file to be read in will do including in the middle of earlier recordings Partial Pascal keeps you informed of what it's reading If it comes to a header with a name other than the one you've asked for, Partial Pascal shows you the name it has found and continues searching the tape

The Partial Pascal programming package includes full-screen editor, co-resident compiler, run time interpreter, utility library, example programs and user manual 18K RAM required $30 postpaid from

Semper Software
1569 Brittany Court Wheaton, IL 60187

Try using the grid to find the numerical codes (in hexadecimal) for these instructions.

LD C,(HL)	4E
LD E,4	1E 04
LD (18418),A	32 F2 47

You will notice that the grid only gives you the *first* part of the instruction As usual, you still have to translate the rest of the instruction into the appropriate numbers, whether decimal or hexadecimal, for the second or third byte of the instruction

Everything You Ever Wanted to Know About...

Page 5 has, in compact form, a tremendous amount of information about each 8-bit load instruction

For example, the first line tells us all about loading information from one register into another, (The small "r" means any one of the regular registers Look down at the bottom of the page under "Notes" and you will find a note that tells you just that.)

What does it say about the instruction? First, it shows us exactly what happens when the Z80 processor performs the instruction The arrow shows you where the number moves—to the first register (the destination) from the second register (the source) This simple system for indicating what happens can be very useful, especially if you are exploring an instruction that you have never used before

Next, it shows what happens to the flags when the Z80 performs the instruction. Some instructions will always raise, or *set*, a particular flag, others will always lower, or *reset*, the flag Some instructions will affect a flag depending on the result of the instruction

In this case, none of the 8-bit load instructions affects any of the flags That is what the black dot under each column means—that the flag is not affected The note on "Flag Notation" at the bottom of the page shows the effect on the flag

· = Not affected
1 = Always raised (or set)
0 = Always lowered (or reset)
0 = Effect depends on the result of the instruction

Some of the flags are really useful only for specialized, complicated tasks, but three of them can be very handy We already know about the *zero* flag and the *carry* flag Information about them is in the columns headed with the letters Z and C The zero flag is usually raised (or set—the words mean the same thing) when an addition or other operation results in a 0 The carry flag is raised when the result is greater than 255, or less than 0 However, some instructions will affect these flags in ways you might not expect Do not try to guess—always check the chart to be sure of how flags are affected

When you translate an instruction into numerical code, the "No. of bytes" column reminds you of how many extra bytes are required.

One other useful flag is the *sign* flag If the result is greater than 127 i e., if bit 7 is one instead of zero, the S flag will often be set. (Remember, bit 7 is sometimes used as a "sign" bit, that is where the name of the flag comes from)

Like most Z80 manuals, the guide uses the words "set" and "reset" rather than "raised" and "lowered" for flags These are jargon words but they are the jargon the guide uses. So to avoid confusion, we will use them too Just remember "set" means the flag is up, "reset" means the flag is down

One other column in this chart can be very useful the column headed "No. of bytes" This column reminds you of exactly how many bytes long each type of instruction should be When you translate an instruction into numerical code, that can be a big help It may keep you from forgetting how many extra bytes some instructions require

Once over Lightly

Pages 6 and 7 give the same information for load instructions for the register pairs and double-sized registers—the "16-Bit Load Group."

Pages 8 and 9 cover the exchange, transfer, and search groups. Though there is no space to explain these instructions in detail right now, you might want to take the time to look at the "symbolic operation" column You may be able to figure out what each of these instructions does just from the grids and charts, now that you know how to read them

Page 10 covers 8-bit addition and subtraction and the logical instructions Page 11 includes some general arithmetic instructions such as NEG, along with a few special control instructions In general, you should not use the "Miscellaneous CPU Control" instructions on your Sinclair or Timex computer most of them stand a good chance of crashing the system

Then the guide continues through the rest of the instructions 16-bit arithmetic, jump, rotate and shift; test, set, and reset, call, and so on The "input" and "output" instructions are really only useful if you are a hardware designer, the same is true of the information about the "interrupt structure" of the Z80

At the very back of the guide is a *very* technical section—all about three special chips that are often used with the Z80 processor. If you have some background in digital circuit design, you may find this section interesting and useful If not, do not worry. You do not need to know anything about the PIO, CTC, or SIO to program your ZX80, ZX81, or TS1000

Coming Attractions

Next time, we will return to some really practical machine code programming we will look at a way to draw circles on your screen Maybe you think it takes the SIN and COS functions in Basic to draw a circle, but, as we will see, you can do it better and faster with a simple algorithm that draws a circle *four times as fast* as SIN and COS—and that is still in Basic. In machine code, it is *really* fast

If you have comments or questions about machine code programming or something is not quite clear, let me hear from you Be sure to send along a stamped, self-addressed envelope if you need a reply

Search and Replace Routines

Jon Passler

Search your program and replace all occurrences of one character or keyword with another.

The routines in Figure 1 and Figure 2 are designed to search through a program and replace all occurrences of one character or keyword with another (keywords are represented with one byte, too). The routines can all be entered directly from the keyboard into a 1 REM line.

All characters and most keywords can be accessed directly by pressing the correct key or combination of keys except the keywords printed *above* the key outlines. To enter these, first enter the keyword THEN, followed by the desired keyword. Then backtrack with shift 5, hit DELETE to erase THEN, press shift 8, and continue. Figure 1 is a print-out of each routine, while Figure 2 has the assembly, along with the characters used for each instruction.

Figure 1.

```
VERSION I  ROUTINE A
  1 REM 5RNDYU,7■C,YP■ 08/■7■
C■777, SCROLL TAN
  2 LET A$=
  3 LET A=L
```

Two versions of the routines are given; version I is for replacing any character, such as the letter A or keyword PRINT with another letter or keyword. Version II is for replacing only string variable names. Version I will do this too, but version II will not replace numeric variable identifiers with the same letter as the string variable identifier.

Three routine variations—A, B, and C—are included for each version. The A routines are for one-time type use. The character to be replaced (marked with a single asterisk in the Figure 2 assembly), and its replacement (double asterisk), have to be entered manually, either by editing line 1 or POKEing (addresses are given in the footnotes).

Figure 2.

```
version I
routine A
LD HL,16542     5, inv 2, RND
LD A,59         V, V
ADD B,B         gre 5
INC HL          ?
CP (HL)         inv Y
JRZ,+9          C, gre 8
LD A, *         Y, A
CP (HL)         inv Y
JRNZ,+11        4, inv Y
LD (HL), **     U, B
JR +7           /, gre 5
INC HL          ?
CP (HL)         inv Y
JRZ +5          C, gre 5
INC HL          ?
INC HL          ?
INC HL          ?
JR,-35          /, SCROLL
RET             TAN
```

* Character to be replaced (16525)
** Replacement character (16530)

```
routine B
LD HL,16557     5, inv H  RND
(Same as Routine A)
LD A,(16530)    U, inv    RND
PUSH AF         PRINT
LD A,(16525)    U, inv B, RND
LD (16530),A    M, inv  , RND
POP AF          LET
LD (16525),A    M, inv B, RND
RET             TAN
```

```
routine C
LD HL,16540     5, inv 0, RND
(Same as Routine A)
LD A,(16530)    U, inv >  RND
LD (16525),A    M  inv 9, RND
RET             TAN
```

```
version II
routine A
LD HL,16549     5, inv 9, RND
LD A 59         V, V
ADD B B         gre "
INC HL          ?
CP (HL)         inv Y
JRZ,+16         C, (
LD A, *         Y, A
CP (HL)         inv Y
JRNZ,+18        4, >
INC HL          ?
LD A,9          Y, 9
CP (HL)         inv Y
DEC HL          F
JRNZ +11        4, gre 5
LD (HL), **     B, 9
JR,+6           /  gre ?
INC HL          ?
CP (HL)         inv Y
JRZ,+5          C  gre 5
INC HL          ?
INC HL          ?
INC HL          ?
JR,-37          /  STEP
RET             TAN
```

* String variable letter to be replaced (16525)
** Replacement letter (16537)

```
routine B
LD HL,16566     5, inv 0, RND
(Same as Routine A)
LD A,(16537)    U, inv :  RND
PUSH AF         PRINT
LD A,(16525)    U  inv B  RND
LD (16537),A    M  inv :  RND
POP AF          LET
LD (16525),A    M, inv B  RND
RET             TAN
```

```
routine C
LD HL,16538     5  inv F  RND
(Same as Routine A)
LD A,(16537)    U  inv :  RND
LD (16525),A    M  inv 9, RND
RET             TAN
```

Jon T Passler 344 Cabot St. Beverly MA 01915

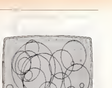
The B routine will flip-flop two characters each time it is run, without requiring any editing. If you want a routine to change all PRINTs to LPRINTs, and vice-versa, then routine B (version I) is best for this application.

The C routines keep track of the last character changed *to*, but the replacement character must be entered manually by editing, or by POKEing. If you want to interchange A$, B$, and C$ in your program (for differently formatted files, for instance), then add a Basic routine that will let you pick one and will POKE its code into address 16537 (for version II). When routine C is run, it will change whatever string variable is presently used to the one specified. If you use numeric variables A, B, and C to hold the format limits, you can use routine C, version I, to change string and numeric identifiers simultaneously.

I will run through version I quickly (version II is basically the same, except as noted). Enter 1 REM from version I, routine A—use Figure 2 and check yourself with Figure 1. Enter lines 2 and 3 as well.

All routines are run by entering RAND USR 16514 in the immediate mode (without a line number). Do this, and the A$ in line 2 and A in line 3 should now read B$ and B. Edit line 1 and change it to routine B (also change lines 2 and 3 back to A$ and A). Try running routine B several times, stopping with A$ and A. Now edit line 1 and enter routine C, and run it. Note that now both 16525 and 16530 hold the letter B. If routine C is run again, it will not change any characters. Enter POKE 16530, CODE "C" and run routine C again. Now enter LET L$="E" and POKE 16530, CODE L$, and run the routine once more. Version II is the same as version I, except the replacement character is held at address 16537 instead of 16530.

"Beats the heck out of bridge, doesn't it?"



Bubbles

Bill Russell

The weekly meeting of our Central PA Timex/Sinclair Users Group was coming up quickly, and I had promised a demonstration of my new TS2068 computer. I needed a special program fast to show off some of its new features.

As I slipped the TS2068 from the box, I wondered if it was free of defects and if it would operate properly for the demonstration. With little time for testing, it occurred to me that the demonstration program could also serve as a test for the proper operation of at least some of the features of the computer. I hooked it up and switched it on. Everything seemed to work fine, so far. Thumbing through the manual, I made notes for my program.

The completed program "Bubbles" is found in Listing 1. The screen display in Figure 1 shows a typical printout. The desired results were achieved: a program for the TS2068 that is short and simple, that provides some degree of testing for proper operation, and that demonstrates several of the new features of the TS2068. (And, yes, the computer has performed flawlessly since it was first switched on.)

What does the program do? It generates circles of random size placed randomly about the TV screen. Screen border color is changed randomly each time a circle is drawn. All the while, random musical notes of random duration are produced. At random times, the screen clears and a new pattern of circles is created. The total effect is arresting and is guaranteed to draw smiles.

"Bubbles" demonstrates some features that are not found on the TS1000, such as the BEEP and CIRCLE commands, sound, and high resolution graphics. But this little program barely scratches the surface of the sound, graphic, and color capabilities of this remarkable new computer. Books soon to be released by the Timex Computer Corporation and other sources, along with articles in *SYNC* will describe in depth how to use these more sophisticated capabilities.

We found one more use for "Bubbles." When displayed on a color TV in the corner of, e.g., a space age living or working area, it provides a kind of High Tech Environmental Art, a moving, abstract, tonal computer painting.

```
  2 BORDER RND*7
 10 LET x=RND*250: LET y=RND*17
5
 20 LET z=RND*100
 24 BEEP RND*.7,z-40
 25 IF z>x OR z>y OR z+x>=255
OR z+y>=175 THEN GO TO 10
 30 CIRCLE x,y,z
 35 IF z>50 AND z<60 THEN CLS
 40 GO TO 2
```

Line Notes:

2 Sets the screen border color.

10 Generates a value, z, for a circle radius.

24 Generates a duration for the BEEP commands and uses the value of z to select the pitch of the beep.

25 Checks to see if a circle centered at x,y with radius z will fit within the limits of the TV screen. If the circle fits, the next line is executed. If the circle will not fit, the program jumps back to line 10 to generate new circle coordinates and radius.

30 Draws the circle.

35 Checks to see if the current value for z falls between 50 and 60. If it does, then the screen is cleared.

40 Returns to the beginning of the program.

Bill Russell, RD 1, Box 539, Centre Hall, PA 16828

Watch Where You Are Going

Sharon Zardetto Aker

In many games you need to know if something is already in the space where you are about to print another character. Whether you are firing a missile at a target, having PacMan gobble dots, or checking for a dead end in a maze, the method is the same: calculate the new position for your moving object, check that spot on the screen *before* moving the character, and continue the program according to what is found in that spot. Regardless of which ZX/TS computer you are using, the theory is the same; the techniques, however, are quite different.

Display Files

The display file is the portion of the computer's memory that keeps track of what is stored on the screen. The location of the TS1000/1500 and ZX81 display file depends on the length of your Basic program. The address of the *print position* in the display file is stored in addresses 16398-16399. You can find this by

PEEK 16398+256*PEEK 16399)

On the TS2068 the display file has a permanent address, but it is arranged in such a manner that simply PEEKing it will not do you any good. The SCREEN$ command, however, is made especially for checking the display

SCREEN$(10,15)

will return the string that is in the tenth row, fifteenth column. SCREEN$ does have limitations: not all characters will be "recognized." However, the asterisk is one that registers, so it will be used in our game program.

A Game Application

Let's set up a simple "Gobbler" game to illustrate how to "look ahead" on the screen.

Both programs begin in the same way: 20 asterisks are printed on the screen at random positions and a "G", representing the gobbler, is printed at the center. The cursor control keys (unshifted) will move the gobbler around; line 50 waits for a key to be pressed.

The new gobbler position must be calculated without changing the variables

Listing 1. Gobbler for the TS1000.

```
10 FOR A=1 TO 20
15 LET R=INT (RND*22)
20 LET C=INT (RND*32)
25 PRINT AT R,C;"*"
30 NEXT A
35 LET R=10
40 LET C=15
45 PRINT AT R,C;"G"
50 IF INKEY$="" THEN GOTO 50
55 LET M$=INKEY$
60 LET NR=R+(1 AND M$="6")-(1
AND M$="7")
65 LET NC=C+(1 AND M$="8")-(1
AND M$="5")
70 PRINT AT NR,NC;
75 IF PEEK (PEEK 16398+256*PEE
K 16399)=23 THEN GOSUB 100
80 PRINT AT R,C;" "
85 LET R=NR
90 LET C=NC
95 GOTO 45
100 FOR A=1 TO 10
105 FAST
110 SLOW
115 NEXT A
120 RETURN
```

Listing 2. Gobbler for the TS2068.

```
10 FOR a=1 TO 20
20 LET r=INT (RND*22): LET c=I
NT (RND*32)
30 PRINT AT r,c;"*": NEXT a
40 LET r=10: LET c=15
45 PRINT AT r,c;"G"
50 IF INKEY$="" THEN GO TO 50
55 LET m$=INKEY$
60 LET nr=r+(1 AND m$="6")-(1
AND m$="7")
65 LET nc=c+(1 AND m$="8")-(1
AND m$="5")
70 PRINT AT r,c;" ": IF SCREEN
$ (nr,nc)="*" THEN BEEP .05,0
80 LET r=nr: LET c=nc: GO TO 4
5
```

that store its current position because we will need those coordinates to print a space at the old position to give the illusion of motion. Variables for the new row and column are introduced at lines 60 and 65; the new positions are calculated using the logical AND instead of IF-THENs to make the program shorter. The column number increases if 8 is pressed, decreases if 5 is pressed, and so on.

A Parting of the Ways

When we check what is printed in the newly-calculated position, the two programs lose their similarity.

For the TS1000, it is necessary to move the print position without actually printing anything; this is done in line 70. Once the print position is placed at the new spot, we PEEK there. If the character code for an asterisk (23) is found at the new position, line 75 sends the program to a gobbling subroutine which, by alternating FAST and SLOW, makes the screen shake.

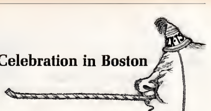
The TS2068 version is much simpler: if the SCREEN$ of the new position is an asterisk, a BEEP is sounded.

The programs differ in another respect. On the TS2068 the gobbler is erased before the check is made. This is simply because, for the best effect, the gobbler should disappear, the tone should sound, and the gobbler reappear, in that order.

Once the gobbler has been erased (line 80 in Listing 1, line 70 in Listing 2), it is reprinted in the new spot by changing the regular row/column values to equal the new row/column values and looping back to line 45.

These programs are simply for demonstration purposes. They need a lot of work to function as real games, e.g., some error trapping to keep the row/column numbers from reaching off-screen values.

Sharon Zardetto Aker, 70 Courtland Dr., Sussex, NJ 07461



A ZX/TS Celebration in Boston

Dennis Krill

On October 22, 1983, the Boston Computer Society's Sinclair-Timex User Group celebrated their Second Anniversary at the Boston Park Plaza Hotel

And what a celebration it was, with exhibits, seminars, and new Timex/Sinclair product announcements! Group Director Sue Mahoney and her staff (Will Stackman, Jack Hodgson, Jeff Parker, Beth Elliot, Bob Masters) organized what must be considered the premier Timex/Sinclair event in the U S Participants included Timex officials, Sinclair Ltd. representatives; vendors of hardware, software, books and publications, and users Exhibitors came from San Francisco to London and Atlanta to Toronto Top that off with over 1000 visitors, and you can imagine the level of excitement. Events ranging from exhibitions to seminars overflowed the meeting rooms.

Timex will make available both technical and supportive information to vendors who want to support the new machines with hardware and software.

Message from Sinclair Research Ltd.

Maggy Bruzelius, Executive Vice-President, Sinclair Research Ltd , assured attendees that, although Clive Sinclair has taken an interest in pocket-sized television (to be available in the U S in early 1984) and the development of an electric car, be is also maintaining his desire to provide computers and equipment at the lowest cost. She also stated that the research division will continue to investigate and evaluate computer related possibilities and applications.

U.S. Connection

A superb demonstration and discussion conducted by Dan Ross, Vice-president, Timex Computer Corporation, created interest and excitement!

The audience was first treated to an analog/digital "Timex" watcb generated and operated by the new Timex Sinclair 2068 color computer The display on an RGB driven monitor was clear, crisp, and bright (the value of a monitor over a common TV was obvious)

His next statements must be considered as among tbe most important assurances offered by any computer manufacturer Mr Ross told the assemblage that the Timex Corporation will be an "open" company! That is, both technical and supportive information will be made available to those vendors who desire to support the new machines with hardware, software, and related peripherals This is definitely good news for vendors and users alike He emphasized that the Timex Computer Corporation recognizes the need for third party vendors in order to be a success in the volatile personal computer industry By providing as much aid and information as possible, users, vendors, and tbe corporation itself will derive the maximum benefit possible.

New Products

While the Timex Sinclair 1500 (black and wbite video with on-board 16K RAM) is in reality an updated TS1000, programs and upcoming peripherals are going to be compatible. This is certainly most important to those of us who have invested in previous software and hardware. It also means that all TS1000 users will be able to keep pace witb advancing technology. Another important aspect of such compatibility is that a 16K RAM pack can be used with the TS1500 to provide a 32K machine. The TS1500 will list at under $80

A new bardware device, referred to as the "T" dock, plugs into the RAM port of either machine to allow the use of cartridge-based software. Thus, program loading will be instantaneous. Again, the technical aspects will be made available to third party vendors.

And speaking of program loading, it seems that Timex sympathizes with the problems we have all experienced. Sit down and get ready for this next new product. Timex will make available (probably as you read this) a Digital Tape Recorder! The expected list price will be under $50 I can't wait to give my system the gift it needs.

For those wbo want a mass storage device, the micro-drive is nearing completion Details of its operation were not available, but I can tell you that it will be about the size of two stacked cassette cases The interface will plug into the RAM port and will not add to the "wire clutter" we now suffer

Interest in a Modem is high With a target price of $100, the Timex-developed unit seems worth waiting for since it will be available by mid-'84 It, too, will operate on all Timex machines

Dennis J Krill, 10 Continental Courts, RD 4, Bellefonte, PA 16823

For the Timex Sinclair 2068, an item referred to by Mr. Ross as the Chameleon is being investigated. Once plugged in, the Chameleon will let Spectrum software run on the new Timex machine, thereby opening to users a large amount of readily available software.

The TS2068 has Atari compatible joystick ports on both sides of the keyboard. While some may scoff at this capability (and I, for one, do not care to see computers used for "games"), the joystick will provide easier operation of one program in particular, VU-3D. Being schooled in architecture with an interest in graphics, I have been tracking CAD-CAM development for the past 10 years. In that time I have "played" with systems costing \$50,000 to \$120,000 (someone else's, of course!) With the TS2068 and VU-3D program, I can generate similar graphics and investigations in my own computer room for 1/200 the cost!

Let me explain the display generated by the computer and Mr. Ross. Using a preproduction program, a 3-dimensional outline of a glass was called to the screen. Then it was rotated (the display passed a fraction of a second between movements) to the desired position. A "remove hidden lines" routine was called, and, before our eyes, all the lines behind the front surface began to disappear. Then a light source position was defined and entered. When the shade command was called, bright surfaces and shadows appeared. The result was impressive. Combining this program with a graphics pad will provide any user with an excellent system. Everyone should have this program in his file, if only to impress friends.

Mr. Ross's final demonstration was a Scrabble game with a 12,000 word vocabulary. While the game has four-player capability, it was more interesting watching the computer play against itself. Although it is still in the planning stages (de-bugging, licensing, etc.), my wife and I eagerly await its release.

Judging from these presentations, the future of the Timex Computer Corporation certainly looks bright. Providing third party vendors with technical information will broaden support for Timex's computer line. With more people producing, it appears to me that the Timex computers (1000, 1500, 2068) will continue to grow with advancing technology.

Seminars and Workshops

There was information available for anyone, from the beginner to the expert.

A gratifying demonstration was presented by Dr. Gregory Coffin, Director Urban Schools Collaborative, Northeastern University. Eight students and their teacher, Judy Fields, from the Timilty Middle School used TS1500s to demonstrate the teaching of computer literacy. The students at first seemed nervous and quite aware of the audience. That changed, however, when they entered their names to the screen. From that point on they worked diligently and quietly.

Brent Jeffries discussed applications of data storage and word processor programs.

Bill Russell's discussion of the ins and outs of starting your own cottage industry provided much-needed information for prospective hardware and software producers.

Bob Masters demonstrated the uses and ramifications of *VU-Calc* and *The Organizer* programs.

Joanna Grammon, a booking agent, told of her application of data filing programs in conjunction with her work.

Dr. Sandra Hutchins' seminar on computer literacy was directed at educators.

Paul McGarry presented interesting biomedical instrumentation applications using the TS1000.

Dr. Bill Corba discussed an integrated curriculum package (K-12) based on the TS1000.

Alger Salt demonstrated his uses of the Votem and the Hunter board.

Dennis Krill gave an overview of word processing programs, complete with loading problems.

Rita Carr ran a program on an TS2068 written by Bob Orrfelt. It demonstrated the sound capabilities of the machine by playing "Maple Leaf Rag."

All day long the Boston Computer Society Sinclair-Timex User Group presented exhibits including a history of Sinclair and Timex computers, a machine language clinic, a chess exhibit and tournament, and a demonstration of how to write and play adventure games.

The vendors' room was bristling with activity one hour before the Celebration began! On display were the latest software, peripherals, books, publications, and speech recognition systems. A robot controlled from a TS1000 piqued the interest of young and old alike. And a surplus computer parts distributor was on hand to provide the "tinkerers" with food for thought.

From the amount of activity provided, it is obvious that much information was made available to those who attended. But more importantly, everyone was given the opportunity to meet and interact with people who, prior to this event, were only names or voices on the phone.

Sue Mahoney and her staff not only made the Second Anniversary of the Boston Computer Society Sinclair-Timex Group a true "Celebration," but also demonstrated the overall value of third party vendors and user groups. ■

Las Computadoras de Costa Rica

Ken Berggren

Costa Rica has been described as an island of sanity in tumultuous Central America. With strife in Nicaragua and civil war in El Salvador to the north and the United States' uneasy ally Panama to the south, Costa Rica maintains stability and close ties to the U.S. Although the Costa Rican economy has taken a fall in the past few years, it appears that even that has stabilized. All in all, Costa Rica is a delightful place to visit and a nice place to live, too.

I came to Costa Rica a little over a year ago to teach science at Colegio Metodista (Methodist High School). This year I began giving classes in computer programming as well. I use the Sinclair simply because it is cheap. The economy is on an even keel, but it is still not moving very fast.

Even though the Sinclair is cheap, it is an excellent tool for teaching Basic programming and about computers in general. The graphics are not too impressive, but they make up for that by being easy to use. SLOW is very slow but for learning that can be a boon. The students can almost watch each step being performed on the screen instead of instantly seeing the final result. One of my students who has had some experience with other computers thinks that the Sinclair is much better because you do not have to type out P-R-I-N-T. All of the commands are already written for you.

The one serious problem with the Sinclair, RAM pack crashes, worried me a great deal as I was planning the course. But the RAMstrap solved the problem. I have not had a RAM pack crash while using the RAMstrap.

SLOW is a boon for learning; students can almost watch each step as it is performed on the screen.

I start the course with simple graphics using the PRINT AT and PLOT commands. Then I introduce variables and how the computer does arithmetic. Most of my studentss are eighth graders so I only mention the more advanced mathematical functions. I do not spend the time to explain them at that level. Next, I present the IF statement. Among other things, the students use it to build some simple controlled loops. Later, the FOR-NEXT loop is presented as a shortcut to do the same thing. I also give them strings, arrays, subroutines, etc.

My method comes straight from the "How to Solve It" series in *Creative Computing* magazine. I find that it is very effective. At the begining of the class I explain some new command or technique. Then I present a problem for the students to solve using what they have just been given. After that, I am there to answer their questions and help them if they get stuck.

A typical "help" session goes like this:

"Mr. Berggren, this computer is stupid! Look at what it is doing!"

"It's only doing exactly what you told it to do. Let's look at the program step by step."

And after a few minutes they will say, "Oh, of course, I forgot to tell him to do that."

I have not mentioned that English is the second language of most of my students. That causes no real problems. Still, it is interesting to explain the NEW command, have them watch its effect, and then see in their notes, "NEW—para borrar todo" (to erase everything). Although the class is in English, I encourage them to use whatever is easier for them in their notes.

Besides my computer classes, I am also Vice-President of the Computer Club of Costa Rica. (The title sounds more impressive than it is.) Most of the members have Sinclairs, again because they are cheap. However, there are more expensive computers represented, too. In the club meetings we share information about computers, hardware, software, books, etc.. The club has just began a project to build a Sinclair controlled weather station. Our present goal is to gather and process data on temperature, humidity, and barometric pressure. Later we could add wind speed and direction if we want it.

I have limited this discussion to the Sinclair computers because they are what I am familiar with and what this magazine is about. However, the big boys like MAI, IBM, Wang, and others are active here in Costa Rica as well. If Costa Rica is typical at all, then computer technology is moving rapidly into the third world countries as well as the first. But I would like to thank Clive Sinclair and his company for making such a neat little package and for putting it at a price that even the economically strapped can afford. ▪

Ken Berggren, c/o Colegio Metodista, Apdo 931, San Jose, Costa Rica

Highway Robbery

Sheldon Maloff

Highway Robbery is a fast-paced machine code driving game specifically designed to fit into a basic 1K machine. With no modifications it can also be run on 2K and greater.

Loading the Machine Code

Entering the program will be done in a series of six steps, and, if you are a good typist, possibly only three steps. Follow the instructions exactly and do not do anything that is not mentioned, especially LISTing. LISTing a program when you are not supposed to could be hazardous. SAVE frequently.

Step 1

Type in Listing 2, the hex loader. There are 294 X's in line 10 so it is best

Listing 1. The hex codes.

```
HEX CODES FOR "HIGHWAY ROBBERY"

0D 00 AD A8 AC AD 8C A8
28 22 27 B4 A7 A7 AA 27
2E 28 0D 00 00 00 00 00
20 00 00 00 00 5C 5C 5C
20 00 20 22 22 22 28 3A
34 40 30 A8 40 00 00 3E
00 07 D7 10 FC 01 00 00
ED FF 0C 40 13 01 00 40
ED 22 12 21 53 A5 42 2A
0C 40 21 04 00 10 22 A2
42 38 24 32 A4 42 3E 22
2A A0 40 23 23 60 E1 23
01 1F 00 ED 00 E0 35 A0
EA 3D 00 F2 0A A2 40 7E
32 A7 42 35 22 3A A0 42
4F C0 07 C0 07 47 C0 27
C8 07 00 01 C8 0C 5F 30
A8 40 0E 00 FF 05 00 C0
13 17 01 32 23 E1 10 F7
3C 0A A5 40 11 01 60 10
3D 00 FC 3A 34 40 CA 2F
3E 00 01 02 C0 05 77 01
2F FF 12 F2 25 20 F0 3A
A7 40 FE 02 00 12 FE 5D
20 2D 2A A5 40 34 7E FE
A8 22 24 38 5C 2A 15 F2
0E 00 0A A2 40 E5 01 36
19 06 05 13 20 71 E0 71
E8 12 F0 3A 20 40 FE F7
C8 70 EE 22 4F 18 E3 11
01 00 0A A0 40 35 22 3A
22 42 FC FD 22 07 FE 2F
2A 14 C3 D0 40 3A A4 40
F8 00 20 F8 19 22 A2 40
3C 32 A4 42 12 2C 3A A4
42 3E 38 E0 32 A4 40 E0
00 2A A2 42 10 00

TOTAL = 30861
```

Sheldon Maloff, 102-432 Huntsville Cres. N.W. Calgary, Alberta, Canada T2K 5E1

With a bird's eye view of an 8-lane highway your Greater-Than-Mobile is cruising to scoop up dollars from the road.

Listing 2. The hex loader.

```
10 REM XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX
20 FOR A=16514 TO 16807
30 SCROLL
40 INPUT A$
50 POKE A,(CODE A$-28)*16+CODE A$(2)-28
60 PRINT A,,A$
70 NEXT A
```

to use the FAST mode. RUN the program and in response to each "" at the bottom of the screen type in a hex code from Listing 1. Type the codes off the list from left to right and top to bottom line-by-line. When typing the hex codes *proceed slowly!* Beware of 8's and B's which look a lot alike.

Should you ever enter a code incorrectly press ENTER, instead of typing the next code in the list, and the program will stop with 3/50. In the immediate mode type

LET A=A-1

Then type

GOTO 40

to restart the program and give you another chance to enter the code. When the code has been entered, it will be scrolled off the top of the screen, so you may not see it, but it has been changed. Continue entering the rest of the hex codes.

When all the codes have been entered and the program has finished with report 0/70, make several SAVEs of the program. In case something should go wrong in upcoming steps you will not have to re-enter the codes. *Do not list* the program.

Step 2

Without deleting the current listing, type in Listing 3, which will overwrite the previous program. RUN the program. The program is finished with report 0/70. If the number at the top of the screen is *exactly* the same as the TOTAL noted at the bottom of Listing 1,

Listing 3. Validity check.

```
20 POKE 16512,2
30 LET T=0
40 FOR A=16514 TO 16807
50 LET T=T+PEEK A
60 NEXT A
70 PRINT T
```

proceed to Step 3. If not, you made a mistake in entering the hex codes and should proceed to Step 4.

Step 3

Without deleting Listing 3, type in Listing 4. *Now* delete Listing 3. (The reason for this is that if line 20 is deleted without Listing 3 present, line 0 will attempt to list. In a 1K machine there is not enough room on the screen for the line to appear and the program will crash.) Make several SAVEs of the program. After this is done, the game is ready to be RUN. Use SLOW mode

Listing 4. The Basic program.

```
1 RAND USR 16559
2 CLS
3 GOTO 1
```

Step 4

Enter Listing 5 and the one currently in memory will be overwritten. Do not worry about deleting line 10 with the one in Listing 5, the machine code was moved to line 0 by Listing 3. RUN the program. It will display 8 bytes of pro-

Listing 5. Code displayer.

```
10 FOR A=16514 TO 16887 STEP 8
20 FOR O=0 TO 7
30 LET D=PEEK (A+O)
40 PRINT A+O," ";CHR$ (INT (D/16)+28)+CHR$ ((D/16-INT (D/16))*16+28)
50 NEXT O
65 PAUSE 50000
70 CLS
80 NEXT A
```

```
Label     Instruction          Comment
CAR-POS                        ;position of car on screen.
CAR-LIN                        ;line of screen car is on.
SCORE                          ;position of ones digit of score.
CAR-STAT                       ;status of current car position.
SEED                           ;random number generator seed.

 START    LD A,(FRAMES)        ;place a number in the SEED holder.
          LD (SEED),A

 BLACK    LD B,90h             ;set up counter to black in top
          LD A,80h              nine lines of screen.
          RST 10h              ;black in two characters at a time
          RST 10h               and continue until 9 lines are
          DJNZ,BLACK            blackened.

          LD BC,32             ;set up counter
          LD DE,(D-FILE)        and destination
          INC DE                and source pointers.
          LD HL,16514
          LDIR                 ;transfer game title to screen.

          DEC DE               ;save address representing the ones
          LD (SCORE),DE         digit of the on-screen score.

          LD HL,(D-FILE)       ;set up location of car on the
          LD DE,148             screen and its starting line
          ADD HL,DE             and save data.
          LD (CAR-POS),HL
          LD A,4
          LD (CAR-LIN),A

SCROLL    LD A,8               ;set number of lines to scroll left.
          LD HL,(SCORE)        ;create the destination in HL
          INC HL                and then transfer it to DE
  LINE    INC HL
          PUSH HL
          POP DE
          INC HL               ;produce source in HL registers.
          LD BC,31             ;set transfer count.
          LDIR                 ;transfer the line left once.
          EX DE,HL             ;make the character in column 31
          LD (HL),80h           a black space
          EX DE,HL
          DEC A                ;repeat until all lines have been
          JR NZ,LINE            scrolled.

          LD HL,(CAR-POS)      ;get the car's position
          LD A,(HL)             and get the status of the position.
          LD (CAR-STAT),A      ;save the status
          LD (HL),92h           and display the car.
          LD A,(SEED)          ;get the last seed and proceed
          LD C,A                to generate a new seed
          SLA A                 and a random number.
          SLA A
          LD B,A
          SLA A
          SLA A
          ADD A,B
          ADD A,C
          ADD A,0Dh
          LD E,A
          LD (SEED),A          ;save the new seed.
          LD C,B               ;proceed to divide random number
          XOR A                 by B.
          LD B,8
DIVIDE    RL E
          RLA
          SUB C
          JR NC,+3
```

```
Label    Instruction          Comment
         ADD A,C
         DJNZ,DIVIDE          ;the remainder is now in A which is
         INC A                 pointer to a row (1 to 8).
         LD HL,(SCORE)        ;get the location of where to plot
         LD DE,21h             a dollar or a brick wall
POINT    ADD HL,DE             in the HL pair.
         DEC A
         JR NZ,POINT

         LD A,(FRAMES)        ;using the TV frames counter check
         BIT 5,A               whether to plot a brick wall
         LD A,88h              or a dollar bill
         JR Z,BRICK
         ADD A,
BRICK    LD (HL),A             and display the choice.%

TIME     LD BC,FF01h          ;load the timer
         DJNZ,TIME             and do not continue until
         DEC C                 the timer runs out.
         JR NZ,TIME

         LD A, (CAR-STAT)     ;analyze the status and if hit
         CP 88h                a brick wall go to explosion
         JR Z,BOOM             routine.
         CP 8Dh               ;if it is not a dollar then
         JR NZ,KEY             advance to keyboard check.
SCOREUP  LD HL,(SCORE)        ;get the score pointer and add one.
         INC (HL)
         LD A,(HL)            ;if the score had not overflowed then
         CP A6h                do a keyboard check.
         JR NZ,KEY
         LD (HL),9Ch          ;if overflowed then make digit a zero.
         DEC HL               ;point to next higher digit
         JR SCOREUP            and repeat

BOOM     LD C,85h             ;load blowup symbol in C register.
EXPLODE  LD HL,(CAR-POS)      ;load both HL and DE with
         PUSH HL               position of car.
         POP DE
         LD (HL),12h          ;display normal car (not inversed).
         LD B,5               ;explosion depth counter set.
DEPTH    INC DE               ;produce an explosion of symbols
         DEC HL                to the depth requested.
         LD (HL),C
         EX DE,HL
         LD (HL),C
         EX DE,HL
         DJNZ,DEPTH
         LD A,(LAST-K)        ;get keyboard character
         CP F7h                and if 1,2,3,4, or 5 key is pressed
         RET Z                 start a new game.
         LD A,C               ;redo explosion with a new symbol.
         XOR 80h;LD C,A
         JR EXPLODE

KEY      LD DE,21h            ;set movement value.
         LD HL,(CAR-POS)      ;turn off car symbol by erasing it
         LD (HL),80h           from the screen.
         LD A,(LAST-K)        ;get the keyboard status
         CP FDh                and if A,S,D,F, or G keys pressed
         JR Z,DOWN             then move car down a lane.
         CP BFh               ;if H,J,K,L, or ENTER keys are pressed
         JR Z,UP               then move car up a lane.
GAME     JP SCROLL            ;start game loop over.
DOWN     LD A,(CAR-LIN)       ;get current car line and if already
         CP 8                  on the bottom lane do not
         JR Z,GAME             move it any lower.
         ADD HL,DE            ;move car down one lane
         LD (CAR-POS),HL       and save the new location
         INC A                 and the new lane.
SAVE     LD (CAR-LIN),A
         JR GAME
UP       LD A,(CAR-LIN)       ;get current car lane and if already
         DEC A                 on the top lane, do not move it.
         JR Z,GAME
         LD (CAR-LIN),A       ;save new car lane.
         SBC HL,DE            ;move car up one lane
         LD (CAR-POS),HL       and save new location.
         JR SAVE
```

gram and their addresses at a time. These should be checked against those in hex listing. Any that differ should be noted on a piece of paper as to the address of the offending code and the correct hex code that should be placed at that address. After 8 bytes are displayed and checked, press the ENTER key and 8 more bytes will be displayed. (This program will also display the bytes at addresses 16808 and 16809, before ending with report 0/80. These bytes can be ignored as the machine code program ends at address 16807.)

Step 5

Enter Listing 6, overwriting the listing already in memory. This program, when RUN, will allow changing offending hex codes. In response to ADDRESS, type

Listing 6. Code changer.

```
 10 CLS
 20 PRINT "ADDRESS ";
 30 INPUT A
 40 PRINT A
 50 PRINT "HEX ",
 60 INPUT H$
 70 PRINT H$
 80 POKE A,(CODE H$-28)*16+CODE
H$(2)-28
 90 PAUSE 32000
100 GOTO 10
```

the address (in decimal) of the incorrect code. Then, in response to HEX, type the correct hex code that should be at this address. If there are more codes to change, press the ENTER key, if not, press BREAK.

Step 6

In the immediate mode type

POKE 16546,118

and

POKE 16547,118

Delete lines 10, 80, 90, and 100 of Listing 6. (Note that when you are deleting the lines, all that will appear on the screen is a REMark statement in line 0 giving the title of the program.) When the lines have been eliminated in the immediate mode type

POKE 16546

and

POKE 16547

proceed to Step 2.

Playing the Game

You are presented with a bird's eye view of an eight-lane unmarked highway and your car, a Greater Than Mobile (inverse >). Due to the speed at which you are cruising, everything appears to be moving past you. To move your car up a lane, press either the A, S, D, F, or G keys. To move down a lane, press H, J, K, L, or ENTER keys. (The reason for the vast array of keys is due to a limited keyboard checking routine which checks only for a key-press in a specific row, rather than a specific key. This method saves memory and allows the advantage of choosing where to place your fingers.)

The object of the game is to move your car from lane to lane to pick up dollar bills lying on the road. Collecting a dollar simply requires you to move your car over top of one. You get one point, or should I say dollar, for each bill collected. Beware of the brick walls (graphic H). If you hit one of those, your car explodes and the game ends. Pressing either the 1, 2, 3, 4, or 5 keys will start the game again.

Variations

Changes to the basic game should be made only when you have succeeded in creating a working version of the program.

Those who live in the United Kingdom may want to enter Listing 7 in the immediate mode to change the dollar into pound sterling.

Listing 7. Pound Sterling modification.

```
POKE 16514,140   POKE 16528,140
POKE 16542,140   POKE 16687,4
POKE 16706,140
```

Currently your car is an inverse greater than symbol. This can be changed to any symbol you wish by using Listing 8 in the immediate mode. The codes 148 and 20 will change the car into an equals sign.

Presently your car is located in the middle of the screen. This is accomplished by the value held at address 16589, which contains a 148. A lower value will move the car closer to the left edge of the screen giving more time to plan your "path of movement." A higher value will provide a more challenging game by moving the car closer to the right edge of the television screen. POKE values at this location must range from 138 to 159, or the program may crash or fail to operate properly.

The speed of the car is set by the value at address 166690. Presently it is set at 15. If you like speeding, decreasing this value will increase the speed of the car. If you feel the game is too fast as it is, a larger value will slow it down.

In order to make modifications after the game has started, you must get back

Listing 8. Car symbol modification.

```
POKE 16530,(CODE OF THE INVERSE
CAR SYMBOL)

POKE 16730,(CODE OF THE NORMAL
CAR SYMBOL)
```

into Basic. To do this, wait until you lose. As soon as you press the key to restart, press the BREAK key. Then make the modifications you want.

For those who want to study the inner workings of the program a disassembly is provided.

Word Processing on the ZX/TS Computers *Sharon Zardetto Aker*

Word processing is the buzzword these days, *the* application for home computers Are you one of the few who would admit befuddlement as to just what the phrase encompasses? Are you one of the many who are wondering just what kind of word processing you can do with your computer? Read on!

La Program

There is a joke making the rounds about the perfect word processor. readily available, cheap, easy to use—a pencil A pencil, however, is to a word processor as a butter knife is to a Cuisinart.

Word processing is a little more than getting sentences from your mind to paper—a job for which a pencil is adequate However, the computer power behind a word processor makes it vastly superior to even the best typewriter Insertions, deletions, and other corrections are made on-screen, instead of on paper, ensuring perfectly clean hard copy

You can automatically format text; switch the order of already written words, sentences, and paragraphs, check the text for every occurrence of a certain word or phrase; or do any of a host of other nifty little routines. Multiple copies at the touch of a key and storage of a master text on convenient magnetic media are only some of the additional benefits of a word processing package

Sharon Zardetto Aker 30 Courtland Dr , Sussex NJ 07461

Uses and Limitations

The limitations of the word processing programs developed for the ZX/TS computers are largely due to its limited memory space the more features included in a program, the less room there is for text

There are also limited uses for word processing on your ZX/TS computer with the TS2040 or ZX printer You would not want to use it for writing the great American novel Besides the lack of memory space, no publisher would accept a 32-column printout, and, of course, the membrane keyboard would probably drive you crazy along about the third chapter

Despite its limitations, the ZX/TS computer is perfectly adequate for a

Figure 1. The Feature Chart.

	Text Capacity (K)	LineLength	Full Screen Text	Word-Wrap	Full Screen Edit	BlockDelete	GlobalEdit	StringSearch	BlockMove	Tabs	Line-Feed	Justification	Double-Space	NewPage	
Hi Res Word Processor	5		•	•	•			•	•		•				Lower case on screen and to printer; 64-character lines
Text II	6	32	•	•		•		•							64-character lines inverse printing
Texter	12	30													
Textwriter 1000	9	32	•	•	•	•			•	•	•		•		block insert
TP16	10	32	•												
VU Write	11	30	•						•						tab markers
Word Sinc II	8	42	•	•	•	•	•	•	•	•	•	•		•	lower case special charecters
ZText	8	31	•				•		•	•	•	•		•	quickload/page no /YANK entries during printing
Z-Wryter	7	30	•		•	•							•	•	triple space block insert
	Capacity		Entry		Edit					Format					Special

number of word processing applications, e.g., informal notes, newsletters with perfect 32-character columns, and personal records. All you need to do is choose the program that will best suit your needs.

The Feature Chart

The chart in Figure 1 compares eight of the word processing packages currently available on fourteen features in four basic categories.

Capacity

The line length figures in the chart refer to the maximum number of characters you can place on one line. Many programs allow you to specify shorter text lines. The text capacities are approximations for comparison purposes—ten thousand characters make about 300 lines. All the capacities are based on 16K RAM; most of the programs are easily modified for computers with larger memories.

Text Entry

Most of the programs have *full-screen text*: everything you type is displayed on the screen, with text disappearing only as the screen is filled.

Word wrap is an important feature. It enables you to continue typing even though you have reached the end of the display line. Without word wrap, anything you key in after the end of the display line will be lost.

Edit

All programs allow you to backspace and correct the current line.

Full-screen edit allows for moving the cursor up and down on the display in order to make corrections. Programs lacking this option require an entire line to be retyped in order to make corrections on any other than the current line.

Block delete lets you designate a specific portion of the text (longer than a line) to be erased from memory.

A *global edit* tells the computer to replace any designated string every time it occurs with another specified string. The computer can check your text for every occurrence of a particular word or phrase with a *string search* feature.

A sophisticated feature that I was surprised to find in a 2X/TS package is *block move*. This lets you shift large portions of the text from one place to another, so you can rearrange sentences or paragraphs without retyping anything.

Format

Being able to set *tabs* that will automatically advance the cursor position is a basic feature that did not show up very often. *Line-feed*, which allows you to start a new line on your display/printout, is not needed in the programs that allow you to enter only one line at a time, but it is a handy feature in the programs with word wrap.

Justification aligns the right hand margin of the text. The *double space option* allows you to choose that format for the final printing. *New page* is similar to line feed, it usually makes the printer advance a number of lines before continuing with the text.

The Report Card

Figure 2 gives a report card evaluation for each program. Except for the overall grade, the programs were rated for how well they did in different areas, not how many areas they did things in.

Documentation was graded on clarity and thoroughness, neatness did not count, except for subliminal psychological effects. Main and submenus were also judged for clarity, and for ease of use.

Programs with word wrap rated the highest in the ease of entry category; those that require character counting and an ENTER for each line, and/or entry of line numbers, were graded progressively lower.

In regard to the entry speed, most of the programs work in machine code or FAST mode for keying in the characters of the text, some, however, take longer to store each line in memory, or to process changes made during an edit mode.

For the error handling grade, programs were judged both on how they handled "dangerous" commands such as "erase" (Do you have a chance to confirm the command?) and on what happens if the wrong key is pressed—does the program break, is all the text lost, how easily can you get back into the program, etc.

The "over-all" grade takes into account both the marks received on the report card and which features are available in the program.

The Reviews

Text II

Text II has both some of the best and worst features of all the software reviewed.

First, the bad news entering text requires first entering a program line number, then a REM statement. Although a line number is not necessary for every display line, the documentation suggests you use one for at least every sentence, since editing is done by line number. The program is designed to BREAK in almost every spot you would expect a pause, requiring a RUN or CONT to go on. The main menu does not offer a "type" or "write" option, and it took me almost twenty minutes to get started entering text, that gives you an idea of the documentation. Printing is slow, with a pause between every text line. Despite the fact that there is a machine code program loaded separately, there is not as much room for text in *Text II* as in the other programs.

The good news is that the machine code routines for inverse printing and program line renumbering are usable with your own programs. In addition to its unique inverse display/printing option, there is another feature that may be *Text II*'s saving grace. You can set 64-character lines and have the first half of each line printed, then the second half of each line. This gives you two pages that can be taped together onto standard sized paper, with sentences reading right across the page.

Texter

Texter is a basically good program with two small problems. Unfortunately, one affects the ease of entry and one affects the ease of editing, however, they are both easily remedied with a little program change if you are up to it, and the author is busy on a revision.

The current problems are lack of full-screen text and line number prompts. Each line disappears as it is ENTERed, making text entry a little confusing. Editing and continued text entry are done by specifying a line number, but lack of prompts (what was the last line entered?) make it slow going.

Textwriter 1000

The report card just about says it all for *Textwriter 1000*. Some features you might wish for, especially a global edit, and, to a lesser extent, string search, were given up to make more room for text. Keep an eye out for Fingerle's TS2068 version, because this one is terrific, and the next should be very special, when he has a little more memory space.

TP16

The main problem *TP16* is that after each line is typed, you must enter a number code for it to either be entered into memory or edited. As if that is not enough to break up the typing flow, there is at least a three-second delay for *each* line to be entered into memory. To make matters worse, if you forget to enter the number code and enter the next line of text instead (a mistake I made repeatedly) the whole program crashes.

Vu-Write

Vu-Write is another that requires a line number entered for each line of text, although just pressing ENTER will print the next line number. Pressing ENTER twice in succession, however, changes the line number without inserting a blank text line, and later editing becomes very confusing. Using a character line of more than 28 also makes editing difficult, because the line numbers will no longer appear on tha screen. There are markers to help you align your text by spacing over to the tab mark.

Word Sinc II

I already had a rave review prepared for *Word Sinc* when this revision became available. *Word Sinc II* is extraordinary.

Lower case—yes, *lower case*—letters from your Sinclair computer to your Sinclair printer, as well as apostrophes, exclamation points, arrows, backslash, and other special characters, all in a 42-character line.

Absolutely flicker-free entry of the text is handled by machine code. Characters repeat at an adjustable speed.

Your display, of course, cannot show lower case or special characters, but symbols have been chosen with special care and common sense. The apostrophe, for instance, is the graphics character often used as the Sinclair default apostrophe, brackets are the upside-down and reversed-L graphics character. The four arrows are entered from the cursor control keys, and so on. As a result of the thought and care that went into the design, confusion has been kept to a minimum.

The documentation could be a little clearer, but the on-screen prompts are fine and, anyway, experimentation is a pleasure with a program like this.

ZText

ZText's nicest feature is Mindware's Quickload process. It takes about two minutes to load this program, compared

	Documentation	Menus	EntryEase	EntrySpeed	EditEase	ErrorHandling	Over-all	Price	Supplier
Hi Res Word Processor	B		A	A+	A+	A	A+	CC $25 00 +$1 s&h	Nissan Elmaleh, 5100 Highbridge St, 53D Fayetteville, NY 13066
Text II	D	B	F	B	B	B	C	CC $20 00	Peak Software, PO Box 8005, Suite 23, Bouldor, CO 80306
Texter	B	B	B	A	B	B	C	CC $6 95 List $2 95	Barlog Software, 401 N Geyser Rd, Kirkwood, MO 63122
Textwriter 1000	A+	A	A+	A	A+	A	A	CC $11 95 List $5 95	Robert Fingerle, 39639 Embarcadero Ter, Fremont, CA 94538
TP16	B	B	F	C	C	F	D	CC $9 95	Marantha, PO Box 759, Mableton, GA 30059.
VU Write	A+	A	C	A	B	B	B	CC $14 95 +$1 s&h	Syncmaster, Rt #1, Box 122, Oak Ridge, NC 27310
Word Sinc II	B	A	A	A+	A	A	A+	CC $20 00	Gesang Associates, PO Box 452, Randallstown, MD 21133
Z Text	B	A	A	B	B	A	B	CC $19 95 + s&h	Mindware, 15 Tech Cir, Natick, MA 01760
Z-Wryter	A	A	B	A	A+	A	B	CC $12 95 + $1 25 s&h	Robotoc, 59 C St, Perrysburg, OH 43551

to an average of eight minutes for the others. Although the instructions say that anything over 31 characters on a line will result in the excess being lost, I found that was not the case; perhaps the program was revised and the documentation was not On the other hand, the string search feature worked only for the line at which the cursor was placed, instead of throughout the text.

Printing is very slow, with about a five second delay between lines, but there are some very nice features "Yank" pulls the most recently deleted line out of storage and places it at the current cursor position—a limited block move. Variables can be inserted into the text and words entered from the keyboard in their places during the printing process Other extras include page numbering and setting the depth of top margins

Z-Wryter

Z-Wryter received a final grade of B because it does not have word wrap, and because it has fewer edit and format options than the highest rated programs. Everything it does, however, it does extremely well. The cursor in the edit mode has repeat movement, and it moves rapidly, pages as well as lines can be inserted and deleted The documentation and menus are well-written and recovery from a BREAK is simple. It is an excellent program despite its shortcomings.

Hi Res Word Processor

Hi Res Word Processor lacks the fancier features of word processing and has a few small drawbacks, however, its main feature—lower case letters on the screen as well as to the printer—pushes it right up to an A+ rating

Text entry is flicker-free, and the text cursor is a thin underline. The keyboard defaults to lower case, with capitals accessed by shifting into graphics mode While editing, you can jump forward or backward a page (screen), and insert or delete entire lines A print option not included on the review copy of the program that will be available on the final product is the 64-character line feature.

The minor drawbacks involve the arrangement of text into individual lines Although there is word-wrap, there is no automatic justification available, so you have to be careful not to split words at the end of lines While you are editing, inserting too many characters means losing the end of a line, and the text does not close up if you delete portions of a line

Although this is a rather expensive program, it does include the hi-res program that sells separately for twenty dollars, and may be well worth the investment.

Keyboard Alternatives

Lawrence A. Kelly

The keyboard converts the character on the key into a code of 8 binary bits—one of the 256 unique combinations of 1's and 0's.

Introduction

Although the membrane keyboard was a major factor in Sir Clive Sinclair's producing the first personal computer under $200, a full keyboard seems to be the peripheral of most interest to the ZX/TS user. However, it is impossible to do touch-typing on this keyboard.

The Anatomy of The Keyboard

The membrane keyboard is composed of two sheets of plastic imprinted with a gold foil circuit. The sheets are sealed together in such a way that a small air bubble keeps the foils separated. When the key area is pressed, the air bubble is compressed and the gold foils make contact closing the switch. When the pressure is released, the compressed bubble restores the separation of the foil.

The ZX/TS keyboard is a diode matrix. When a key switch is closed, +5V is applied through one of five resistors to the anode side of one of eight diodes (a diode conducts in only one direction). The diodes connect directly to 8 address lines which the Z80 CPU scans via a Sinclair ROM routine which Dr. Ian Logan (the original disassembler of the Sinclair ROM routines) calls KSCAN. The five resistors connected to 5 volts comprise the columns of the matrix and the diodes the row (Figure 1). This 5 × 8 array accounts for the 40 ZX/TS keys.

The Language of the Keyboard

The purpose of any keyboard is to be an interface or an interpreter. It converts the letter, number, or character found on the key into a code of 8 binary bits. The number of possible unique combinations of 1's and 0's in 8 bits is 256.

Appendix A of the Manual (either version) explains how the 256 possibilities, numbered from 0 to 255, are used in the Sinclair system. The left most column is the decimal value of the code. This can be very helpful in Basic programming since, if you "PRINT CHR$(N)" where N is that number in the left column, you will print the corresponding charecter to the screen.

The column labeled "Hex", for hexadecimal, i.e., 16 (hex=6, decimal=10). Hex numbers are useful here because 4 binary bits of 0 and 1 can code for 16 unique things, i.e., $2^4 = 16$. If you follow the table to the end, you will see that it only takes a combination of two hexadecimal numbers to cover the whole range of 256 binary codes. The figures 00 = 00000000 to FF = 11111111b cover all the codes used by the ZX/TS computers.

To get more than 256 codes Zilog cheats a little. The hexadecimal numbers (they are not letters) CB and ED are prefixed and another set of instructions apply. The jibberish in the remaining three column are the mnemonics (short names to assist memory) for the assembly language instructions. The reason for going into this much detail on how the keyboard interprets what is typed on it, is that later we will discuss ASCII encoded keyboards.

It is, therefore, helpful here to explain what the keyboard does in encoding some outside world languages into binary information which the computer can handle. Two main codes are in use today in the world of computers ASCII (say "as-key"), the American Standard Code for Information Interchange, and EBCDIC (say "eb-see-dic"), Extended Binary Coded Decimal Interchange Code. These codes cover the letters (26 capitals, 26 lower case), numbers, and punctuation characters that one usually encounters on a keyboard plus the control characters used in data transmissions as originally established for teletype machines. ASCII does this on 7 bits, allowing the 8th to be free for error checking. In ASCII there are $2^7 = 128$ possible items coded.

The bottom line of this discussion is that there are two major codes and the one in Appendix A is neither. Sinclair uses a special code which must be translated to one of the other codes when trying to communicate with other computers. For this reason keyboards that are already encoded are difficult to connect to the ZX/TS computers.

Keyboard Alternatives

A First Approach. Keyboard Overlays

One of the major problems with the membrane keyboard is thet you do not feel a key moving under your finger. As a result you can tell only by the screen result that you have connected with a key. The flat membrane feels the same whether you are pressing on a key pad or on a space between. When I was first using the membrane keyboard I often resorted to using two pencils, pressing the key pads with the rubber erasers. This does not do a lot for the "tactile sensation," but it does give traction and prevent slippage of fat clumsy fingers from the keypad area.

The overlays approach this problem. They are simply plastic sheets with holes bored out over the keypad area. Like braille, these holes can be felt, and the fingers can have the proper key location. These overlays usually come with an adhesive backing which can be peeled off and pressed onto the membrane keyboard.

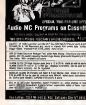
I have seen a number of these advertised in *SYNC* with various shaped holes etc. The first one I recall seeing advertised was from Kopak Creations, 119 Peter St., Union City, N.J. I have used the Kopak overlay at their Union City location and found it to be a step toward touch-typing on the membrane keyboard. The cost of the plastic sheet overlay is usually under $10.

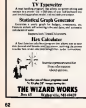
A newer overlay approach, also advertised in *SYNC*, is a "chiclet type" raised key which glues over the membrane keyboard. This innovation is from Filesixty and costs $19.50.

The Second Approach. Beepers

Some people select the ZX/TS computer because of the membrane keyboard. In certain applications it is desirable. For example, kitchen appliances with microprocessors generally employ membrane keyboards to protect the electronics from spills. The IBM 9000 laboratory computer comes standard with the membrane keyboard, and

Lawrence A. Kelly, PO Box 484, Morris Plains, NJ 07950





the IBM PC type keyboard is an option

A tactile sensation is replaced by an audio sensation. Devices for audio feedback are available for the ZX/TS computers.

The Third Approach: The Full Keyboard

This is the approach that most ZX/TS owners find to be the most gratifying. They then feel like they have a real computer when they feel the solid "click" of a key.

If there is a users group available, contact them for names of members who have done full keyboard conversions. Also, have you noticed the additional bonus with your *SYNC* subscription, i.e., the reader service card? Look through the magazine, identify keyboard expansions you might be interested in, and circle their numbers on the card. The advertiser will send additional information directly to you.

Installing a Full Keyboard

There are three ways a full keyboard can be installed.

1) Convert a Surplus Keyboard

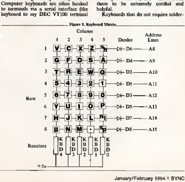
The main advantage of converting a keyboard is that it allows you to select a keyboard that you really like. However, unless you already have the keyboard, or are well versed in electronics as well as computer keyboard interfaces, I would advise leaving this possibility alone. Computer keyboards are often hooked to terminals via a serial interface (the keyboard to my DEC VT100 terminal connects via a banana plug with 3 connectors, transmit, receive, and ground) To connect that keyboard to the ZX/TS the circuit board would have to be rewired to conform with the matrix shown in Figure 1. If a keyboard is connected by a serial interface it is usually already ASCII encoded, as mentioned earlier. This is a hinderance rather than a help. In general, ASCII encoded, separate keyswitches, and Hall effect keys are all problems. The best bet is a matrix type keyboard with a matrix that closely resembles the one shown in Figures 1 and 2.

2) Purchase a surplus keyboard already converted to the ZX/TS matrix

The keyboards are outlined in Table 1. Those from Double H Electronics, LJ H Enterprises, and Kopak fall in this category. Typically these will cost you $10-20 more than a keyboard from a flea market or electronics surplus house. I would say that it is well worth the additional expense to save the rewiring job. You can then get the documentation as well as the support, from the supplier. I have personally installed several keyboards from Double H for myself and friends and have found Herb Hornung to be extremely helpful. I have heard the same thing from people who installed the L J H. keyboard from Leonard Holmberg. The Kopak keyboard is only recently introduced, but I have found them to be extremely cordial and helpful.

Keyboards that do not require solder-

Figure 1. Keyboard Matrix.



ing do require complete removal of the membrane keyboard My experience with this is that, once done, it is difficult to reverse The plastic membranes with the gold foil on them continue out of the keyboard and become the leads which plug into the edge connectors on the circuit board The foil is very delicate, with about one inch/once out, amount of gold foil on them Once pulled out to put in the keyboard connectors it is unlikely that all keys will work upon to re-installing the membrane keyboard

In this regard the keyboards that solder on have an advantage in that both keyboards can be kept operational by soldering the keyboard to the solder side of the circuit board and never touching the membrane keyboard Figure 3 (I include Figure 3 because usually this view is shown from the opposite side of the board) shows the solder side of the board and where the connections should be made on the keyboard matrix. The printed circuit board does not have to be removed from its case.

What I usually have done in connecting Double H keyboards is to connect a 25 pin female connector (the type usually used for an RS232 connection) to the side of the computer case and connect the lines from the circuit board to the connector I then connect the key-

Figure 2. Keyboard layout.
The customary way of seeing the matrix.

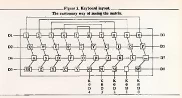
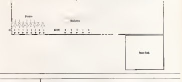


Figure 3. Solder side of computer board.
Dotted lines indicate components located on other side of board.



board to a corresponding male connector. If one then wishes to take advantage of the small compact nature of the ZX/TS, the keyboard can be unplugged and the computer is still functional.

I am not sure if the Byte-Back Keyboard falls in the category of a rewired keyboard, but this company also has a long history of successful devices for the ZX/TS computers. The Jameco keyboard is a rewired keyboard. The version I saw had the case, which is quite large and contains the entire computer and RAM pack and there may still be room for your lunch. The ZX/TS computer becomes Apple in stature in this enclosure. The ZX/TS legends are not on the keys but remain marked on the case.

3) Purchase a keyboard specifically manufactured for the ZX/TS.

Quite a number of keyboards are specifically manufactured for the ZX/TS computers both in the UK and US. Again try to see and use the keyboard. Talk with people who have the keyboard and get their impressions. I have examined directly a number of these keyboards but certainly not all of them. Read any available literature carefully, and, if possible, examine a keyboard before you buy it. I have set up and used the Fuller, DK'tronics, and Memotech keyboards.

The Memotech was certainly the easiest to install. It plugs directly onto the expansion port, little fuss, but with a tiny bit of interference on the screen that was not there with the membrane keyboard only. The secret of the Memotech keyboard is that the diode side of the matrix is contained in the interface that plugs on to the expansion port. The frequency of the Z80 is in the radio frequency range bringing these out from the printed circuit board in an interface always runs the risk of interference with an RF modulated TV display. The interference is minimal and the ease of connecting the Memotech should make it the best keyboard for those with no electronic inclinations.

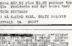
The Fuller and the DK'tronic keyboards both require disassembly of the computer to install in the enclosure. The DK'tronics model is of good quality and has a handy numeric keypad. However, it presents one problem. Because of the way the computer board is mounted in the case, the edge connector does not stick out but rather plugs into a motherboard. The motherboard does not stick out far enough to put the RAM pack on. The recommendation is to disassemble the RAM pack and mount it unfolded on an extension of the motherboard inside the case. This I found to be highly unsatisfactory, but the problem could be solved by using one of the cable

Table 1. Keyboard roundup

Keyboard Suppliers	# Keys	Enclosure	Solder	Numeric Pad	Sinclair Legend	Price	Comments
Double H Electronics 195 Leland San Antonio TX 78242	53	NO	YES	NO	NO	$56.00	Typewriter Style Excellent keyboard action Full Space bar
LJH Enterprises PO Box 6305 Orange CA 92667	41	NO	YES	NO	NO	$85.00	One of the first wired for the Sinclair Full Space bar
JAMECO Electronics 1355 Shoreway Rd Belmont CA 94002	62	YES optional	YES	NO	Legend on Case	$99.95 $59.95 $49.95	+ Case Keyboard only Case only
DK 'Tronics 23 Sussex Rd. Great Yarmouth Norfolk UK	52	YES Computer inside	NO	YES	YES	$89.95	Good quality RAM must be disassembled to attach
K 2 Electronics Design 1990 Varsity Dr Ann Arbor MI 41804	50	YES	N/A	NO	YES	$195.00	Called "Kradle" Comes with 64k RAM Expansion ports
Sinclair Place Indescomp PO Box 2288 Redmond WA 98052	52	YES Computer inside Ext RAM	NO	YES	YES	N/A	Sculptured keys On/Off Switch
"Compact" Sinclair Place PO Box 2288 Redmond WA 98052	56	YES Computer inside Metal	NO	YES	YES Color Codes	$84.80	Contact SP for info
Research Applications Prod 4561 Paloma Lane Yorba Linda, CA 92686	73	NO	YES	Pad for shifted	YES	$80.00	Contact for info.
Kopak Inc 119 Peter St Jersey City NJ 07087	48	YES Optional	NO	Optional	YES	$99.95	Typewriter Style High quality Full Space bar
E-Z Key Suite 75 STX 711 Southern Artery Quincy, MA 02169	60	YES Two Options	NO	YES	YES	$84.95	Not typewriter but large enter 5" Space bar
Zebra Systems Inc 78-06 Jamaica Ave Dept. B Woodhaven NY 11421	40	YES	NO	NO	YES	$69.95	Contact for info
Suntronics Co. Inc 12621 Crenshaw Blvd Hawthorne CA 90250	41	YES Computer inside	NO	NO	YES	$59.95	Full Space bar
Gladstone Electronics 1585 Kenmore Ave. Buffalo, NY 14217	47	YES Computer inside	NO	NO	YES	$79.95	Full Space bar
Byte-Back Co 1702 Oak Knoll Dr Leesville SC 29070	41	YES For Keys only	N/A	NO	YES	$59.95	Full Space bar
Memotech Corporation 7550 West Yale Ave. Denver CO 80227	41	YES For Keys only	NO Interface on back	NO	YES	$99.95	Buffered Interface Plugs on like RAM Pack
Toptronics Company 4018 A W. Chandler Ave. Santa Ana, CA 92704	N/A	N/A	N/A	N/A	N/A	$54.81	No info. available
KAYDE Electronics Systems The Cottage Dept SY 3 Norfolk NR30, UK	41	YES Optional Computer inside	N/A	NO	YES	$75.90 $55.90 $21.90	Repeat key
Fuller Keyboard Ramex Intl 46945 Van Dyke Utica MI 48087	42	YES Computer inside	NO	NO	YES	$49.95	RAM pack can be easily attached
Synergistic Design PO Box 411023 Chicago IL 60641	59	YES	NO	NO	YES	$85.95	Auto Repeat

RAM pack extenions. The Fuller and Memotch keyboards have a nice feature which puts a shift key right along side the delete key. One can then very easily delete by hitting the two keys simultaneously. My major criticism of these three keyboards is that all the keys are the same size. I am used to using a keyboard with a space bar and a large "RETURN" key.

Table 1 reviews a number of keyboards that have been advertised in various publications. Examine some of these before you buy.



The Thurnall System

Lawrence A. Kelly

The Thurnal Modular System. I/O Port $34, Motherboard $30, 8-Way Indicator Unit $24, 8-way Switch Unit. $24, 8-way A/D Converter $45.75, Transistor Driver $18. Thurnal Electronics, c/o Sinclair Place, PO Box 2288, Redmond, WA 98052

The Thurnal Electronics Input/Output Port is a parallel interface, using the Z80 PIO developed by Zilog for the Z80 CPU.

The Bus

As the general public becomes progressively more "computer literate," the topic of conversation at many cocktail parties drifts to personal computers. These conversations are usually liberally modified and amplified with "buzz" words and phrases (one only need mention these, it is not necessary to know what they mean at cocktail parties.)

One such phrase is, "Is it expandable?" As a ZX/TS owner you may validly answer, "Yes, my computer is expandable." Though it does not have a Versabus, Multibus, Unibus, or S-100 bus (some of the more famous micro/minicomputer buses), it does have the expansion port in the rear. This may be called the "Sinclair bus." You usually attach the RAM pack at this point.

A "bus" in computer terminology should not bring to mind a sleek racing dog (Greyhound), but rather the collective lines that connect the various inner parts of the computer. The bus is subdivided into an Address, Data, and Control bus. In our 8-bit, Zilog Z80 based ZX/TS, there are 8 Data lines, 16 Address lines, 13 Control lines, and another 3 lines for +5V, ground, and the clock signal. This gives a total of 40 pins which go into the Z80 computer (CPU) chip. These lines, as well as +9V (unregulated from the power supply) and a memory line, can be seen labeled in the chapter on machine code in your manual (p. 124 of the Timex edition).

The Thurnal Electronics (TE) devices "expand" from this edge connector.

The TE I/O Port

The TE I/O Port is the first part to connect to the computer, and it is literally the portal through which everything passes.

There are two main ways that data can be passed in and out of a computer. The first is the way a program is loaded from the tape recorder, i.e., over one wire with the bits queued up one behind the other, parading single file. This is known as serial transmission. A typical serial interface is the RS232 standard. The other wey of moving data in and out of the computer is on the bus, with the data marching in, shoulder to shoulder, 8 bits wide. This is known as parallel transmission. A parallel standard is the IEEE 488 (sometimes called the Hewlett Packard bus).

The TE I/O Port (I/O = Input/Output) is a parallel interface. The interface is accomplished via an integrated circuit developed by Zilog (designer of the Z80 CPU) to be a parallel interface for this microprocessor. It is called a Z80 PIO (Parallel I/O). From the outside the chip looks identical to the Z80 CPU. The inside is quite different though it has the same lines. The PIO is designed for receiving parallel data from the CPU and outputting it to a peripheral.

There are two separate ports, termed A and B, which have 8 data lines plus "handshaking" (control lines). The TE box has a connector which is identical to the RAM connector which connects directly to the computer. A memory pack is not necessary to operate the I/O port, but, if you will be using one, the computer edge connector for the RAM pack, continues out the rear of the TE box.

The PIO is fully "decoded." What this means to you is that the use of the TE expansions will not interfere with any other devices you intend to attach to the computer such as 64K RAM packs, printers, serial interfaces, modems, etc.

However, because the port is decoded you will need a very short machine code program to access the ports. The documentation from TE offers you an option here, a stepwise entry of the necessary codes into a Basic REM statement which is clearly explained with absolutely no mention of machine code. This is guaranteed to operate the port. They also have a section which completely explains the machine code routine if you are interested.

The connector to the ports has all the lines to port A on the top of the circuit board and the lines to port B on the bot-

Lawrence A. Kelly, PO Box 484, Morris Plains, NJ 07950

tom Any of the TE devices can be connected singly to this edge connector (one of the first things I connected directly here was the TE joystick

If more than one of the TE devices is to be used, a motherboard is available The motherboard will connect up to 4 of the TE devices at the same time (since the ZX/TS is not a multitasking computer, you will be able to address them only one at a time The motherboard contains no electronic devices, it is just an extension of the bus or wiring, namely, the data and control lines—2 to port A (A1 and A2) and 2 to port B (B1 and B2) The proper connecting sequence of the devices and the lines is diagrammed on the cover of the motherboard

The TE Joystick

I set up the TE joystick and entered a rather extensive (16K RAM required) Basic program supplied by TE which constructed a maze that used the joystick to manipulate through the course without crashing into the wall There is very good documentation for the operation of the joystick, so those of you interested in games could quickly adapt one or more of these to your game programs. The *Timeblaster* video game from Caliope Software would be particularly interesting to play with a joystick In this case, since the software is mostly Z80 code, you would have to prevail upon the author to provide you with a version that used the codes you would need for the joystick, instead of for the movement keys which are used

Other TE Devices

Let's take a quick look at five other TE devices The TE Indicator Unit, with 8 LEDs, lights up according to the binary equivalent of what is sent to the port (a good way to learn to count binary) The TE Switch Unit, just like the original micros—the MIT's Altair 80 and the IMSAI 8080—allows you to switch your programs in through the port by setting the binary equivalent of the code on the eight switches (on=1, off=0) and then reading the port (This would be purely academic as far as I am concerned)

Another TE device can do Digital to Analog (D/A) control of devices The TE Transistor Driver controls low voltage electronic devices while the TE Relay Box switches on low current household equipment using line voltages.

In all, the TE devices are a sound investment They are well built and at a reasonable price for Sinclair expansion If you do not have any electronic experience or applications along those lines, you will probably be interested only in the parallel port and the joysticks

resources

"Resources" lists new products and services for Sinclair and Timex/Sinclair users. Suppliers and users are invited to send brief informative, objective descriptions of products and services along with details for ordering to Resources, SYNC, 39 E. Hanover Ave., Morris Plains, NJ 07950

All programs in this listing require the 8K ROM and 16K RAM unless otherwise noted. "CC" indicates cassette format. When a supplier has more than one product listed, the name and address follow the last product.

User Groups

Florida
South Florida Users Group
c/o Bob Pearsall
9220 Fountain
Lakewood FL 33463
Newsletter $10/12 issues

Oklahoma
Timex Sinclair User Group
Billy Casebeer, Pres.
PO Box 372
Oologah, OK 74053

Rhode Island
Ocean State T/S User Group
c/o Bob Dyl
15 Kilburn Ct
Newport, RI 02840
(401) 849 3805

Texas
San Antonio Timex/Sinclair User Group
Jim Houston
414 W. Elsmere
San Antonio TX 78212
(512) 735-2695

User Group Name/ Address Changes

Maryland
Capitol Area Timex/Sinclair Users Group
PO Box 725
Bladensburg, MD 20710
Formerly Prince George's SUG.

Missouri
Timex/Sinclair Users Group of Kansas City
c/o Carl E. Mutch, Pres.
4701 NW Linden Rd
Kansas City MO 64151
(816) 587-8820 Formerly Computer Users Group TS1000 & ZX80/81

New York
Address change
Sinclair Computer User's Society (SINCUS)
PO Box 36
Johnson City, NY 13790

User Group Forming

Pennsylvania
Philadelphia area
Mark F Miller
9573 Walley Ave
Philadelphia, PA 19115
(215) 698-1905 Please send SASE.

Topical
Archaelogy special interest group. Write in English, French, Spanish, or German. Worldwide Archaeological ZX Users Group
c/o Robin Smith
30 Charles St., W. #720
Toronto, Ont
Canada M4Y-1R5

Forth Interest Group
PO Box 1105
San Carlos, CA 94070. Annual dues $15 (U.S.): $27 (foreign). Includes subscription to *Forth Dimensions* (bi-monthly).

Publications

"What Every Computer User Should Know about Static Control." 8-page brochure on nature and cause of electrostatic charges and treatment for static-proofing virtually every surface and object in the computer workplace. For a copy write ACL Inc., 1960 E. Devon Ave. Elk Grove Village IL 60007 (312) 981-9212.

RB Forum. Monthly newsletter published by RB Robot Corp. Free to owners of RB5X Intelligent Robot, other subscriptions $15/yr prepaid. RB Robot Corp., 18301 W. 10th Ave., Suite 310, Golden, CO 80401 (303) 279-5525.

AAMI Journal. Published by American Association of Microcomputer Investors, Inc. Included in membership dues of $49 to provide information on how to use computers to do better in investing. Other membership benefits: comprehensive directory of investment software, free computer programs, discounts on investment software, study guides and disks. Write: American Association of Microcomputer Investors, PO Box 1384, Princeton NJ 08542 (609) 921-6494

Books

McGraw-Hill Dictionary of Scientific and Technical Terms. 1,846 pp. $70. Covers every major field of science, engineering, and technology. Reflects current usage and specialized terminology in 100 different disciplines. McGraw-Hill Book Co., 1221 Ave. of the Americas, New York, NY 10020 (212) 997 3493.

GOSUBS: 100 Program-Building Subroutines in Timex/Sinclair Basic. By Ewin Gaby and Shirley Gaby. $9.95. 176 pp. Designed to provide both novice and experienced programmers with building blocks for constructing, modifying and refining programs on the ZX81, TS1000, TS1500 and, with very few exceptions, the Spectrum and TS2068. McGraw-Hill Book Co., 1221 Ave. of the Americas, New York, NY 10020 (212) 997 2486.

Creative Computing Press (U.K.)

3 new books for the ZX81 and TS1000 from Creative Computing Press available directly from the publisher in U.K. *The Best of SYNC* (£6), *50 Programs for the TS1000* (£5.50), and *The TS1000 Ideabook* (£6). Limited quantities of *Computers for Kids: Sinclair Edition* (£4.25) also available. Add £1 for postage and packing in U.K., £2 to all other countries. Hazel Gordon, 10 Bishops Way, Sutton Coldfield, West Midlands B74 4XU U.K.

Programming Aids

Complete Software Library. Extensions to Basic. Program Name

Reader, Multiple Programs in Memory, Tape Unlocker, Secret ROM Tape Routine, Etchsketch, Character Set Bandana Surprise!, Beep Routine Lower Case for Printer, Phaser Sound Effect, Super Function Plot 16K Two Pass Assembler Hi-res Printer Routine Also Unusual Display Program Name Reader Relocation Program, Oscilloscope with Fantastic Graphics, 16K text editor with full print formatting, 1K text editor, 3 great real-time games, Sines, Flower, Beep Relocation Routine, Unusual User, Keyboard Learning Game Self List Poster Maker Program, Multiple Programs in Memory Relocation Routine, 3D Plane 1K to 16K RAM On TDK tape or equivalent Poster at half price with this package CC $24 95

John Richard Coffey, PO Box 448 Scottsburg, IN 47170

Basic Basic. Learn to write your own computer games; screen display keyboard input, moving graphics, loops, editing, and other programming techniques Program tape workbook and blank tape $17 95 Visa/MC 2-Bit Software, PO Box 2036 Del Mar, CA 92014 (619) 481-3242.

Tape Unlocker. 1K machine code program to end your worries about not having backup tapes Break through the program list and save or print it. Listing $1 plus SASE

Super Save/Load. 4 features SAVEs and LOADs machine code above RAMTOP SAVEs and LOADs data, reads the name of the program on tape makes copies of almost any tape program MC People who have bought *Multiple Programs in Memory* will be able to use this to save all the programs in memory CC $4 95

John Richard Coffey, PO Box 448 Scottsburg IN 47170

RPNZL Programming System. Occupies lower portion of the 16K RAM pack, leaves enough room for equivalent of 12K Basic program in only 6K RAM, typically 15 times faster than Basic Full screen text editor suitable for word processing and a compiler permit structured, symbolic code in a Forth-like stack oriented syntax Accesses dozens of program, data, and text files per tape CC and manual $29 95 See applications elsewhere in this column The Golden Stair, 141A Dore St , San Francisco. CA 94103 (415) 552-1415 $1 50 s&h per order

Word Processors

Word Sinc II. Word processor featuring 42 characters per line lower case letters (unshifted), auto repeat on all keys, full screen editing; automatic up and down scrolling, block move search replace, delete, tab. 20 special characters and more. Room for 8000 characters in 16K RAM, expandable to larger RAMs CC with manual $20 F Hargrave, Site V RR 4, Nanaimo BC, Canada V9R 5X9 Exclusive U.S Distributor: Gessing Associates, PO Box 452, Randallstown, MD 21133

Text Processor. Write, read, edit, print save, and clear text 2000 word capacity Edit in full or split screen mode CC and instructions $12 50 A Rodriguez 1605 Pennsylvania Ave , Miami Beach, FL 33139

Electronics Programs

Electronic Programs for the Hobbyist. Programs cover Ohms law op-amps bipolar power supply design Menu driven Looped programs $9 95, $1 s&h (money order) Rob Wells 3090 Meadowgreen Way, West Jordon, UT 84084

Tax Programs

Fed-Tax Software System. System has 3 separate programs *1040 Long Form, 1040A Short Form 1040EZ Easy Form* and *Schedules A & B* CC and documentation Each program $10 95 plus $1 s&h Toco Technology, PO Box 98, Santa Claus, IN 47579

Math/Sciance

Easy Math. Does math directly type in problem or formula and press ENTER. answer appears on screen 10 single letter variables available for problems like (4+D)/TAN A Each can be an equation itself so systems of equations can be run with a single keystroke Produces columnar output on printer with headings, subtotals, and totals Demo program included CC $9 95 Simulton, Box 894 Lemon Grove, CA 92045

Math by Numbers. Arithmetic tutorial for grades 3-5 Randomly generated problems in addition, subtraction, multiplication, and division of multidigit numbers Enter answer one digit at a time just as on paper allows for carrying, borrowing, and remainders Separate score for each type of problem, extra practice for weakest area CC and instructions $12.95 pp Charles J Bonner, PO Box 9179, Norfolk, VA 23505

Cubic/quadratic Solution. Solution to the cubic and quadratic equation for real coefficients $1 plus large SASE. Igor Alexeff, E.E Dept., University of Tennessee, Knoxville, TN 37916.

Equation Solver. Displays each step in the solution of mathematical equations Works on any level of mathematical proficiency Also solves unknowns CC $8.95 pp. Robert Martin, 15950, Fountain Valley, CA 92708

Lab Program I. Computes CODs, BODs, MLSS/MLVSS/SVI, Total Suspended Solids, Total Solids will run 20 samples in each category, except TSS which runs 34 Formats data in tabular form.

Lab Program II. 2 interactive parts 1) COD, 2) BOD/COD Correlation With user input factors will compute COD, estimated BOD, Sample sizes for BOD analysis, Expected High and Low BOD range, includes HELP screens Auto load feature carries results from one part to another Formats data in tabular form.

Menu driven programs $24.95 each SASE for information Labsoft 1707 King St., Jacksonville FL 32204

'Eat as many lemondops as I do, and you eventually have to pay the price '

Financial Programs

Checks + Balances. Up to 60 data files for storing your checking transactions Automatic number of checks (according to your present system) plenty of room for descriptive info automatic deposit mode, review by number or payee's name auto-save mode saves data CC $8.50 plus $1.50 s&h Computer-Wear Software PO Box 1059 Dept 1D, Riverdale, NY 10471.

IRA Organizer. Maintain a complete permanent record of all your IRA transactions (dates deposits, withdrawals, interest) Subtotals and account balance computed Menu: enter data, display all data, correction, display account summary, display specific transaction, save to tape, copy to printer CC and documentation $16.50 John B. Carson, 11200 Lockwood Dr., Silver Spring, MD 20901

Loan Amortization. Shows breakdown between principal and interest on loan payments use any payment plan give any 3 of 4 factors and the program calculates the fourth Listing $8, CC $15.

Tax Return Organizer. Written by an accountant, helps organize tax records for individuals CC $15

WMJ Data Systems, 4 Butterfly Dr., Hauppauge NY 11788

Rapid Finance. 19 user established categories up to 500 entries made by date, category, amount optional comments Numerical totals maintained for each category Entries may be searched Much MC Info on request CC $10.95 Russell Brewer 26630 Mill Rd., Frazeysburg, OH 43822

Utility Programs

Quickdrive. 17 MC graphic and utility routines to incorporate into your programs, listable, explained, Basic demo Includes CLS/Screenfill, Inverse, Border, Input Check, Graph 1 & 2, Scroll (4 directions), and more CC and manual $12.95 Micro-load, PO Box 1095, Truth or Consequences, NM 87901

Educational Programs

Writing. Input a 100 word sample of text and the program will determine the educational grade level required to understand it An analysis of your text will also be prepared CC $8.95 UAS, PO Box 612, Haddonfield NJ 08033

Math Quiz. Asks questions on grades 1-6 level 7 levels CC $9.95 pp Andy Balderson, 3390 Dent Cres., Prince Albert, Sask., Canada S6V 7H3

Bible Books. Drill and practice in recognizing New and Old Testament books, categorizing them by type (history, letters etc.), spelling of the book names arranging them in order $5 Abbreviated 2K drills $3

Ten Commandments. Drill and practice in completing key words in the Ten Commandments and in recognizing the sequence Available in KJV, RSV, GNB $5

Prodigal Son. 2 programs 1) Presents story in frames for student interaction, variety of questioning techniques. 2) Simulation program for students to reenact the decisions made by the younger son as he leaves his father, travels, squanders his money, nearly starves and then returns in shame to receive his father's forgiveness. Ages 11 up Both programs included $10

Marydel Frohne, St John United Church of Christ, 601 N Highway 83, Bensenville, IL 60106.

Spelling Program (2K RAM) Drills user in self-entered words Fully adjustable word lengths, number of words, length of time to study words, etc Ideal for small children Prompts Documentation and modification instructions In Basic. Listing $1.50 plus SASE. John Osborne Box 579, Oakwood IL 61858.

Home Programs

Auto Diagnostic. Select your problem areas and the computer will perform a decision tree analysis to pinpoint your trouble spots. Many remedies require only simple adjustments. Save on costly auto repair bills CC $9.95 UAS, PO Box 612, Haddonfield, NJ 08033

The Organizer. Inventory program Allows manipulation of items through add, alter destroy, start from scratch

The File Developer. Allows storage of 100 files spread over 4 parts

For more information write E and S Software, PO Box 196, Budd Lake, NJ 07828

Insulator. Figure gains on home energy improvements, determine whether savings justify cost. Listing $2 and SASE. CC: $10 plus $.60 s&h

Biorhythms. Calculate physical, mental, and emotional cycles for any given day or 3 week period. Listing $2 and SASE. CC $10 plus $.60 s&h

MAH Software, 341 W Eagle Lake Dr. Maple Grove, MN 55369. SASE for catalog

Astro-Data. File system with options of the 12 signs of the Zodiac to give information on your character, destiny, future career, and more. CC $10.50 (Canadian; no personal checks or credit cards) J Brunet, 857 St-Honore St Quebec City, Que Canada G1J 2W2

Schedule. Enter appointments and short memos in any order, retrieve them by day, week, month, or year. Holds at least 75 items, up to 250 CC $7.95

Chartmak. Six modes of operation for line, bar, and pie charts CC $9.95

The Golden Stair, 141A Dore St San Francisco CA 94103 (415) 552-1415 $1.50 s&h per order RPNZL application programs

Flying Programs

Flight Planner. For pilots. Plans entire cross country flight, true courses, wind correction, true air speed, groundspeed, etc are all calculated. Maintains data base of airport/VOR locations; will generate checkpoints with enroute VORs. CC and manual $10 pp. Joe Smiley, 8A Anthony Cir., Newark, DE 19702

Amateur Radio

Ham Helper Series. Programs for the Ham "Shack" from data files to QSL printing. In Basic; easy modification. SASE for details. Krug D Pritts, Box 196, Leonardsville, NY 13364

Beam Heading. Calculate bearing, miles, and kilometers from user's city to any location in the world. Prints world grid centered on user's city

Log Book. Provisions for input of all log data, recall from memory by call sign or QTH. Printout can be used for QSL card (QSL format).

DXCC. Keeps all ARRL countries data. View, print, and revise records; printout of countries worked, confirmed, or needed. Single or all band

WAS. Same as *DXCC* but for states

Dupe Sheet. Checks for duplicate contacts before they are called. Keeps contact number displayed for contest use. Prints calls for records

Antenna Design. Design your own cubical quad, yagi beam or inverted Vee antenna for ham or CB radio. Saves hours of research

Each program on CC: $5.95 ea. pp Kentronics PO Box 586, Vernon, AL 35592

Ham-Hacker Series. Morse Code (16K)/2K code. Code practice with built in audio tone. By Gary Fox KC9MR $14.95. Minimuf 3.5 Maximum usable frequency radio propagation program. For DXers. By Sandy Blaize W5TVW $17.95 CE Amp Common emitter amp Design/test program from Syncware By Fred Nachbaur $19.95 Hawg Wild Software PO Box 7668 Little Rock, AR 72217

Food Programs

Microkjeldahl Protein, Determination (S 030). Calculates total nitrogen (dry basis) in MG/100G and percent protein (dry or wet basis) for up to 50 items. Program protected by 4 sets of account numbers and passwords. Menu driven

Moisture-Oven Method (S 031) Helps with tedious computation of moisture content of food and agricultural materials. Accepts raw data from which a table of moisture contents is printed. Up to 50 items.

Program listings: $4 pp. ea.; CC $7 pp. ea. Ako Tech, Dept SL2, 1613 Dayton Rd., West Hyattsville, MD 20783

Graphics

Turtlart. Uses one letter commands: direct the turtle to draw just about any figure desired. Sequences of up to 255 steps can be programmed. RPNZL Application CC $9.95 plus $1.50 s&h. The Golden Stair, 141A Dore St., San Francisco, CA 94103 (415) 552-1415. $1.50 s&h per order

SW Hi Res. High resolution graphics on your TV: software only, no hardware changes needed. Accesses 256 x 174 pixels. Define your own symbols. Includes 10 utilities. CC and instructions: $15. Add $5 for COPY utility; prints Hi Res on TS2040. N Elmaleh, 5100 Highbridge St 53D, Fayetteville, NY 13066

Miscellaneous

Computer Stand Module. Build a tabletop computer stand module to hold a ZX81 or TS1000 computer with 16K RAM pack firmly in place, a ZX printer provision for storing up to 9 tapes with title edge clearly visible Inexpensive to build with simple tools Plans and instructions $2 W G Squiers, 28 West St, Fairfield, ME 04937

Demo Sales Program. For all qualifying TS1000 retailers. Program designed by an advertising director to help you sell more Timex products. For details on you can own this $40 program write Computer-Wear Software, PO Box 1059, Dept 1D Rivardale, NY 10471

Computer Poster (2K RAM) 22 foot by 4 inch computer poster. Very abstract, great conversation starter Poster and fully documented poster program $5 95 program alone $1 plus SASE John Richard Coffey, PO Box 448, Scottsburg, IN 47170

Services

Cassette and disk duplication. Write for details Compdisk, 6224 Oakton St Morton Grove, IL 60053

Games

Golf Handicapper. Compute, save, and update golf stroke handicaps for up to 48 players Menu add players update player handicap, list handicap and scores for a given player, list all players and handicaps, save onto tape, copy to printer, delete player Handicaps computed according to USGA guidelines for 18 hole courses CC and documentation $20 John B Carson, 11200 Lockwood Dr, Silver Spring MD 20901

Pro Golf. 18 holes with all the hazards of real golf sand traps, rough, and 12 clubs to keep you on the fairway, different every time you play

ZX81 Football. Try your luck against the ZX81 or another player, all the features of professional football 42 different plays

Statistics. Keep up-to-date statistics on your favorite team and players Predict scores of upcoming games with amazing accuracy Programs for baseball, hockey, football, and soccer Specify which game you want

Each game on CC $10 95, all three for $21 90 Certified check or money order R D Brennan, 122 The Heights Dr, Don Mills Ont, Canada M3C 1Y3

4-2K Adventures (2K RAM). Tank Track (supreme strategy) 3D caves (can you get out?) City Skipper (arcade action) Zephus Lair (arcade adventure) CC and catalog $10 Adventure Club 2016 Woodhillcrest Rd, Mobile AL 36609

18 Hole Golf Game. Each course, ball, and hole are printed Shows hole number, shot number, and current score of player Courses get progressively harder

Elevator Man. Your job is to operate an elevator in a large hotel If you do not get to each floor quickly enough and get the people into the elevator, they will fall out the hotel windows

Phantom Bombs. Bombs coming from another world are trying to blow up the defense shield of your planet You must stop the bombs by using a large beam which disintegrates them instantly

Air and Sea Attack. Planes and ships are heading in on the attack You must stop them with missiles before they arrive You can control the missiles after they are fired

Any 2 programs for $10, all 4 for $15 pp Original Software, 605 Applehill Rochester MI 48064

2K Game Pack (2K RAM) Key Game, Quick Lander, I Got a Bad Feeling 'bout This, Etchsketch Sines 5 2K Basic listings $1 plus SASE

cen-TIC-pede. MC action high scores multiple levels extra large characters CC $4 95

John Richard Coffey, PO Box 448, Scottsburg IN 47170.

ZX/TS Black Box. 4 or more points are hidden on a grid locate them by sending imaginary beams at them and analyzing the results Challenges your thinking powers One to four players can play against each other Main algorithm and other routines in MC $10 95 plus $1.50 s&h U.S currency Aldebaran Alpha, Box 776, R1, Hudson, PQ, Canada J0P 1H0

The Assassin. The monarchs of the land are being assassinated and the terrorized king has hired you (the Assassin) to eliminate the source of distress Graphic fantasy offers perplexing riddles during real time playing requires strategy

Orgs & Ogres. You are in a party of 10 fighters out to find the gold or kill the dragon You encounter other monsters orges, ogres, kobolds You must choose whether to fight the monster or run To win you must get 1000 gold pieces or slay the dragon

The Dark Empire. You are commander—in-chief of the rebel nobles on the planet Rion Can you liberate your forces from your stronghold and overthrow the tyrant Pyrinx? If you can free Rion, can you help free the Milky Way from the insidious clutches of the Dark Empire?

CC $12 95 each JPR Software, PO Box 4155 Winter Park FL 32793. (305) 646-9125

Wizard's Castle. Fight monsters, hunt treasures solve problems defeat the evil wizard 3D graphics Randomly created multi-level dungeon with over two dozen different types of monsters, treasures, traps, and special items All MC $12 95 pp David Spellman, PO Box 2300, Provo, UT 84603

Hiding. You are an American sub commander searching for a Russian sub hiding in American waters Mixture of graphics and text which acts as documentation Hard game of skill and luck For information write E and S Software, PO Box 196, Budd Lake, NJ 07828.

Gridlock (2K) Four games Tow Truck Tell the driver where to find your stalled car Foggy Sea Cartesian coordinates help you rescue a missing ship Robot Program the robot arms to repack two cartons in the smallest number of moves Easy Plot Displays plot of equations for Y as a function of X

Snake Eyes (2K) Odds On See just how random the random number generator really is Roulette Place your bets you can win this two-player game by chance. Loaded Dice The "house" has fixed this game, figure out how Blackjack Figure the changing odds to beat the computer (dealer) at 21

CC $14 95 each 2-Bit Software, PO Box 2036, Del Mar, CA 92014 (619) 481-3242. Visa/MC.

Mountain Raider. Your mission is to patrol the mountainous regions of Phelld for the demonic craft of the Cyborg Suddenly one swoops down out of the sky. Will it fire on you or is it playing chicken? On not waste your fire power but, if the demon hits you you will walk home Programmable joystick. CC $5 95

The Desperate Herd. Herd roams a hot, torrid plain, grazing on the patchy bush always seeking more until they find it or starve They reproduce only if they get something to eat, but the herd's gluttony leads to a population explosion and famine You control metabolism, energy requirements growth and food value of the vegetation. CC $14 95

Meteor Storm. Fast, arcade-style animation Challenge yourself to command your craft through a storm of interstellar debris. Skills as pilot and gunner stretched to limit CC $7 95

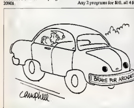
The Golden Stan. 141A Dore St., San Francisco, CA 94103 (415) 552 1415. RPNZL applications. $1.50 s&h per order

3D Black Star. Explosions mushroom around you the rear scanner shows pursuing craft. Increase speed and fire dodge from side to side of the narrow corridor Fast 3D graphics Warp drive Full instrument display By M Sudworth

Damper. A grid runner is needed to shut down the power grid of the city as the space leaches attack A cable cycle is provided, the rest is up to you **Glooper.** A-maze-ing. By P Crane

Pioneer Trail. Western adventure based on historical data 20 levels of play mind game plus shooting Rifle speed uses all keys and is measured against the player's personal average response By Marion Stubbs

Munchees. Beware of the Munchees A board creaks in the time lost corridors A ghostly presence, a rush of wind a forgotten echo whispering around the house and a



following chuckle of ghosts. By A. Laird.

Croaka Crawla. It is not easy being a frog—what with the trucks and logs, crocodiles and turtles. The continual battle for survival. By John Field.

CC. $14.95 each. $2 s&h per order. Quicksilva, 426 W. Nakoma, San Antonio, TX 78216. (512) 340-3684.

New software line for the ZX/TS computers: *Blackjack*, $12.95, *Tic Tac-Toe*, $9.95, *Word Scramble*, $9.95. Free 28 page catalog describes the line. Dynacomp Inc., 1427 Monroe Ave., Rochester, NY 14618. (716) 442-8960.

ZX81 Games Pack. 7 games. 1K: 3-D Battle, City Bomb. 16K: Warp Wars, Snake, Sweet Tooth, Black Holes, Slalom.
Battleships
Alien Attack
JRS Software, 19 Wayside Ave., Worthing, U.K. BN13 3JU.

TS2068 software

Multiple Programs in Memory, Text Editor and Formatter, Three Great Real Time Games, Keyboard Learning Game, Etchsketch, Tape Unlocker, Touch-type Learning Game. CC. $9.95. Substitutions may be made for individual programs. John Richard Coffey, PO Box 448, Scottsburg, IN 47170.

Boards/Interfaces

FDZX1 Interface Board. Buffered access to address, date, and control buses. 6 decoded device codes (expandable to 18). Can be used in automated measurement, data acquisition, and instrument control. School, laboratory, home, industry applications. Series of science experiments to incorporate computers into science courses. Kit $69.95, assembled $99.95, $2 s&h. Visa/MC. Group Technology, PO Box 87, Check, VA 24072. (703) 651-3153.

I/O Experimenter Board. For Spectrum. Do I/O circuit design, 8-bit port, large prototyping area. Bare board with description and application information on how to build a multichannel sound generator. $29.95 (# 2401). Elcomp Publishing, 53 Redrock Ln., Pomona, CA 91766. (714) 623-8314.

ROMs, RAMs, EPROMs

Uvipac EPROM Eraser. Load EPROM into conductive foam pad and insert in unit, usually erases in 5-20 minutes, optical fiber on-indicator. $29.95, same unit with timer $39.95. Airmail shipping $10. Send 1 international reply coupon for information. Ground Control, Alfreda Ave., Hullbridge, Essex, U.K. SS5 6LT.

Instant Load Cartridges. Plug-in software cartridges running programs from a replaceable ROM (also available); load with no wait time, eliminates cassettes. Software now available includes arcade games, party games, finance, utilities, scientific, graphics, and more. Blank cartridge plus any software on ROM: $25, additional ROMs $15 each and plug into reuseable cartridge. SASE for free catalog. Rompak, 8206 Blackburn Ave., Los Angeles, CA 90048.

Disk Drives

Complete Floppy Disk System. Controller (FDC 100). Works with single and double density, single or double sided; supports 2 drives; 40 track double density drive provides 250K storage; new DOS commands integrated into Basic (no USR, PEEK, POKE used), compatible with wide range of Shugart drives. $199. Complete systems Model FDS-101: Controller, cable for 2 drives, Shugart SA-200 Disk Drive, power supply, case for SA 200 with space for second drive. $499. Model FDS 102: Model FDS-101 with the 2nd drive and power supply. $760. $3.50 s&h. For further information write or call. Compusa Corp., 1101 Bristol Rd., Mountainside, NJ 07092. (201) 654-7220.

Display Aids

Video Reverser. Reverses entire screen display. Reduces eye strain; diminishes display distortion. Installed inside case (only 4 wires). Optional switch for switching between normal and reverse. Use all high speed devices for outstanding character definition. $14.95 plus $1 s&h. Sight and Sound Electronics, 1120 Bailey Hill Rd., Eugene, OR 97402.

Keyboards/ Accessories

Contact Lens. Keyboard overlay, made of clear soft non-glare plastic with rectangular holes to accommodate even large fingers. Self adhesive backing; stick to original membrane keyboard. $7.95. Warrent Imports Group, 81 Brookmill Blvd., Unit 80, Agincourt, Ont., Canada M1W 2L5.

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION (Required by 39 U.S.C. 3685)

1. a. Title of Publication: SYNC
   b. Publication No. 02785781
2. Date of filing: October 1, 1983
3. Frequency of issue: Bimonthly
   a. No. of issues published annually: 6
   b. Annual subscription price: $18.00
4. Complete mailing address of known office of publication: 39 E. Hanover Avenue, Morris Plains, NJ 07950
5. Complete mailing address of the headquarters or general business offices of the publisher: 39 E. Hanover Avenue, Morris Plains, NJ 07950
6. Full names and complete mailing address of publisher, editor and managing editor:
   Publisher: None
   Editor: David Ahl, 39 E. Hanover Avenue, Morris Plains, NJ 07950
   Managing Editor: Paul Grosjean, 39 E. Hanover Avenue, Morris Plains, NJ 07950
7. Owner: Ahl Computing Inc., 39 E. Hanover Avenue, Morris Plains, NJ 07950; Ziff Davis Publishing Company, One Park Avenue, New York, NY 10016
8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities: None
10. Extent and Nature of Circulation:

	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total no. copies (net press run)	58,474	85,993
B. Paid circulation		
1. Sales through dealers and carriers, street vendors and counter sales	17,305	27,300
2. Mail subscription	25 877	34 100
C. Total paid circulation (sum of 10B1 and 10B2)	43 182	61 400
D. Free distribution by mail, carrier or other means, samples, complimentary and other free copies	1,926	3,318
E. Total distribution (sum of C and D)	45 108	64,718
F. Copies not distributed		
1. Office use, left over, unaccounted, spoiled after printing	2,056	1 577
2. Return from News Agents	9,310	19,700
G. Total (Sum of E, F1 and 2—should equal net press run shown in A)	56,474	85,993

11. I certify that the statements made by me above are correct and complete.

WILLIAM L. PHILLIPS
Assistant Treasurer

Index to Advertisers

21st CENTURY ELECTRONICS

6813 POLK STREET • GUTTENBERG, N.J. 07093 (201) 869-2616

Hardware

DISK INTERFACE	$189.95
PERTEC DRIVE	$225.00
POWER SUPPLY	$ 89.95
XEROX CASE	$ 49.95
64K RAM	$149.95
16K RAM	$ 49.95
HI RES PACK	$ 99.95
PRINTER N INT	$399.95
EZ-KEY KEYBOARD	$ 64.95
HUNTER BOARD	$ 59.95
FORTH BOARD N DOC.	$ 52.95
SEE THRU INT. COVER	$ 14.95
VU MONITOR	$ 19.95
STARTING FORTH BOOK	$ 22.00
UHF MODULATOR	$ 15.00
BMC MONITER	$109.95
MONITER CONVERSION	$ 15.00
MONITER SWIVEL STAND	$ 39.95
TIMEX/1000	$ 49.95
TIMEX/1500	$ 79.95
TIMEX/2068	$199.95
TIMEX PRINTER	$ 89.95
BYTE BACK MODEM	$149.95

21st CENTURY ELECTRONICS Business Series

A—SALESDATA 2E **$34.95**

16K complete sales storage and analysis report for up to 30 customers for a quarter year. Comes in a book form with a master tape and 2 blanks, and complete forms documentation. A hard copy can be made of all the reports.

B—FILEDATA 2E **$20.95**

16K not only files names, addresses, zip, telephone no., but allows you 2 lines of comment per entry. It contains an alphabetical sort and allows you to change any part of any entry. A hard copy can be made of the complete entry or just the mailing address. All Series 2E programs come with full documentation.

C—PAYDATA 2E **$34.95**

16K complete payroll program for 25 people. Includes hard copy mode for pay stubs, and accountant information, automatic overtime calculations for over 40 hrs., and all federal tax deductions are set in the program. There are also instructions allowing you to set your own state taxes. All Series 2E programs are user friendly.

CIRCLE 72 ON READER SERVICE CARD

Books and Software

THE SINCLAIR ZX81 PROGRAMMING FOR REAL APPLICATIONS	$12.95
ZX-81/TIMEX BASIC AND MACHINE PROGRAMMING	$ 9.95
STARTING FORTH BY LEO BRODIE	$18.00
STARTING FORTH HARD COVER	$22.00
MOTHERSHIP	$16.95
BIORHYTHMS	$14.95
SUPER CHESS	$19.95
MASOGS	$19.95
FINANCIAL MANAGER	$16.95
THE GAMBLER	$14.95
FLIGHT SIMULATOR	$17.95
CHECKBOOK MANAGER	$15.95
THE FROGGER	$17.95
UNDERSTANDING BASIC	$15.00
SALES FILE 16K	$19.95
SALES FILE 64K	$19.95
INVENTORY 16K	$ 9.95
INVENTORY 64K	$ 9.95